



Conception de l'architecture d'un système dirigée par un modèle d'urbanisme fonctionnel

Jacques Simonin

► To cite this version:

Jacques Simonin. Conception de l'architecture d'un système dirigée par un modèle d'urbanisme fonctionnel. Génie logiciel [cs.SE]. Université Rennes 1, 2009. Français. NNT: . tel-00512182

HAL Id: tel-00512182

<https://theses.hal.science/tel-00512182>

Submitted on 27 Aug 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N°ORDRE : 3844
de la thèse

THÈSE

présentée

DEVANT L'UNIVERSITE DE RENNES 1

pour obtenir

le grade de DOCTEUR DE L'UNIVERSITE DE RENNES 1

Mention : INFORMATIQUE

par
Jacques SIMONIN

Équipe d'accueil : Triskell – IRISA

École doctorale : Matisse

Composante universitaire : IFSIC

TITRE DE LA THÈSE :

Conception de l'architecture d'un système dirigée par
un modèle d'urbanisme fonctionnel

Soutenue le 29 janvier 2009 devant la commission d'examen

COMPOSITION DU JURY :

<i>Rapporteurs</i>	:	Colette ROLLAND	
		Alain WEGMANN	
<i>Examineurs</i>	:	Daniel HERMAN	<i>Président du jury</i>
		Yves LE TRAON	
		Francis ALIZON	
		Jean-Marc JEZEQUEL	<i>Directeur de thèse</i>

Remerciements

Je remercie Colette Rolland et Alain Wegmann pour l'intérêt accordé à ma thèse. Leurs remarques constructives me seront très utiles pour la suite de mes travaux. Je remercie Daniel Herman qui m'a fait l'honneur de présider le jury.

Je remercie Jean-Marc Jézéquel d'avoir dirigé ma thèse et de m'avoir fourni une aide précieuse quant à la "transformation" d'un discours d'ingénieur en un discours académique. J'associe à ce remerciement Yves Le Traon pour m'avoir toujours encouragé et pour m'avoir fourni des pistes de recherche que j'ai été très heureux de suivre.

Je remercie Francis Alizon avec qui j'ai partagé mes premières idées sur l'urbanisme et le développement de système. Je le remercie également pour sa grande qualité d'écoute tout au long de cette thèse.

Je remercie la Direction de la Recherche de FT R&D, et plus particulièrement la commission "Thèses et HDR pour IG R&D", de m'avoir fait confiance dans le cadre de mon projet de thèse.

Je remercie Fouz Menaï et Jean-Pierre Deschrevel de FT R&D qui ont permis à ces travaux de démarrer grâce à des problèmes d'architecture bien conçus et à des réflexions pertinentes.

Je remercie Piotr Kołodziejwski, Leszek Kozłowski, Rafał Malczewski, Janusz Kowalczyk et Marek Dabrowski de TP R&D, Katell Henry, Grégoire Dupé, Jamil Chawki et Benoit Hellequin de FT R&D pour leur participation au prototypage des démarches.

Je remercie Olivier Boëffard de l'ENSSAT et Emmanuel Bertin de FT R&D pour les échanges enrichissants que j'ai pu avoir avec eux sur certains aspects de mes travaux.

Je remercie toutes les personnes de l'équipe projet dirigée par Hervé Le Corre et par Alexis Bafcop, ainsi que les personnes de l'unité de recherche et de développement de FT R&D dirigée par Bertrand Nicolas, pour toutes nos interactions sur l'urbanisme des services télécom. Je remercie plus particulièrement Sébastien Poivre qui a enduré un peu plus que les autres l'avancement de ma réflexion. Pour la même raison, je remercie Goulven Furet, mon voisin de bureau, qui a supporté avec bonne humeur mes interrogations en temps réel autour de cette thèse.

J'associe à ces remerciements tous les membres de l'équipe Triskell de l'IRISA que j'ai pu croiser à l'occasion de cette thèse, en particulier, Loïc Lesage pour son efficacité lors de l'organisation de mes différentes venues à Rennes.

Je voudrais enfin remercier Sophie qui m'a accompagné durant ce projet de thèse et qui m'accompagne dans la vie. Elle s'est intéressée à ce domaine de l'architecture, nouveau pour elle, et elle m'a permis de travailler dans les meilleures conditions possibles.

A Sophie, ma femme, et à Marion, ma fille.

Résumé

Résumé

La durabilité d'un système d'information (SI) peut être caractérisée en bonne partie par la réutilisabilité de ses services lors de la réalisation des processus métier de l'entreprise. L'architecture orientée services (SOA) pour un SI nécessite la prise en compte de la stratégie d'entreprise (marketing et évolutions technologiques) dans la conception des services afin d'augmenter la réutilisabilité de services existants. C'est l'objectif de l'architecture d'entreprise (EA), ou de l'urbanisme, qui définit une cible architecturale offrant une durabilité optimum du SI.

Afin d'améliorer la réutilisation de services lors de l'évolution d'un SI, nous proposons d'intégrer une approche dynamique fondée sur l'EA pour toute nouvelle réalisation d'un processus métier de l'entreprise. Le principe de cette approche dynamique est de prendre en compte, lors de chaque usage de services, d'une part, les fonctions et les données manipulées et, d'autre part, les solutions techniques. L'amélioration de la durabilité du SI est évaluée dans notre approche par une mesure d'alignement entre l'architecture des services développés dans le SI et l'EA fonctionnelle de ce SI.

Mots Clés : Système d'Information, durabilité d'un Système d'Information, architecture d'entreprise, urbanisme, architecture orientée service, ingénierie des modèles, ingénierie / architecture dirigée par les modèles, alignement de modèles, mesure d'alignement.

Abstract

The sustainability of an Information System (IS) can be mainly characterized by the services reusability during the implementation of the company business processes. The Service Oriented Architecture (SOA) deployed in an IS requires to take into account the company strategy (marketing and technological evolutions) during the service design. The aim is the existing services reusability improvement. This is the purpose of either the Enterprise Architecture (EA), or the urbanism, to define an architectural target providing an IS optimal sustainability.

In order to improve the service re-use during an IS development, we suggest to integrate a dynamic approach based on EA for any new implementations of a company business process. The dynamic approach principle consists during every services use, on one hand, the functions and the manipulated data and on the other hand, the technical solutions. The IS sustainability improvement is estimated in our approach by an alignment measure between the IS services architecture and the functional EA of this IS.

Keywords: Information System, sustainable Information System, Enterprise Architecture, urbanism, Service Oriented Architecture, model engineering, Model Driven Engineering / Architecture, model alignment, alignment measure.

Table des Matières

Remerciements.....	3
Résumé	5
Table des Matières.....	7
1. Introduction	9
1.1 Contexte	9
1.2 Contribution	11
1.3 Plan du document	11
2. État de l'art.....	13
2.1 Architecture d'entreprise	13
2.1.1 Durabilité d'un SI et réutilisabilité	13
2.1.2 Généralités sur l'EA.....	14
2.1.3 Différents cadres de l'EA.....	17
2.1.4 Un cadre particulier de l'EA : l'urbanisme	18
2.1.5 EA et modélisation	20
2.2 Vues d'architecture.....	23
2.2.1 Méta-modélisation et vues associées au cadre de l'urbanisme	23
2.2.2 Alignement entre la vue métier d'une entreprise et les vues du SI	29
2.3 Développement d'un système	30
2.3.1 Processus de développement UP.....	30
2.3.2 Aspect dynamique d'un système.....	32
2.3.3 Développement d'un système et approche MDA.....	34
2.4 Conclusion.....	35
3. La démarche DA4EA d'urbanisation de la vue fonctionnelle d'un SI	37
3.1 Principes	37
3.1.1 La vue fonctionnelle déduite de la vue métier ou alignée avec la vue métier ?	37
3.1.2 Description d'ensemble de la démarche DA4EA.....	40
3.2 Conception experte des zones et des îlots du PLU fonctionnel	42
3.3 Conception des îlots du PLU fonctionnel	46
3.3.1 Définition du typage d'un îlot fonctionnel	46
3.3.2 Transformation des îlots du PLU fonctionnel contrainte par le typage.....	48
3.3.3 Transformation des voies du PLU fonctionnel contrainte par le typage des îlots	49
3.4 Alignement de la vue métier d'une entreprise et de la vue fonctionnelle d'un SI	53
3.4.1 Définition de l'alignement de la vue fonctionnelle avec la vue métier.....	53
3.4.2 Axiomatisation de l'alignement de la vue fonctionnelle avec la vue métier...	55
3.4.3 Mesure de l'alignement de la vue fonctionnelle avec la vue métier	56
3.5 Conception automatisée des quartiers du PLU fonctionnel.....	60
3.5.1 Approche dynamique de la conception des quartiers.....	60
3.5.2 Pertinence d'un regroupement d'îlots fonctionnels dans une même entité fonctionnelle.....	62
3.5.3 Seuil d'insertion d'un îlot fonctionnel dans une entité fonctionnelle.....	68
3.5.4 Algorithme bactériologique de conception des quartiers fonctionnels à partir des entités fonctionnelles.....	71
3.6 Expérimentation avec le SI supportant le processus métier d'usage d'un service télécom	75
3.6.1 Démarche appliquée à la zone Messagerie	75
3.7 Évaluation de la démarche DA4EA	90
3.7.1 Contexte du prototypage de la démarche DA4EA	91
3.7.2 Bilan du prototypage de la démarche DA4EA	92
3.7.3 Analyse du bilan du prototypage de la démarche DA4EA.....	92

3.7.4	<i>Bilan comparatif avec la zone Messagerie existante</i>	94
3.8	<i>Conclusion</i>	95
4.	La démarche EA4UP de développement de système assisté par l'urbanisme fonctionnel d'un SI	97
4.1	Principes	97
4.1.1	<i>Pourquoi l'urbanisme fonctionnel ?</i>	97
4.1.2	<i>Pourquoi une approche dynamique ?</i>	98
4.1.3	<i>Démarche EA4UP et ingénierie des modèles</i>	100
4.1.4	<i>Description d'ensemble de la démarche EA4UP</i>	105
4.2	Démarche EA4UP fondée sur l'approche dynamique et l'ingénierie des modèles	107
4.2.1	<i>Collecte des exigences</i>	107
4.2.2	<i>Analyse</i>	111
4.2.3	<i>Conception de l'architecture fonctionnelle</i>	121
4.2.4	<i>Conception de l'architecture technique</i>	137
4.2.5	<i>Conception de l'architecture applicative</i>	143
4.2.6	<i>Implémentation</i>	158
4.2.7	<i>Test</i>	160
4.3	Évaluation de la démarche EA4UP	161
4.3.1	<i>Contexte du prototypage de la démarche EA4UP</i>	162
4.3.2	<i>Bilan du prototypage de la démarche EA4UP</i>	163
4.3.3	<i>Analyse du bilan du prototypage de la démarche EA4UP</i>	164
4.4	Conclusion	165
4.4.1	<i>Démarche EA4UP et SOA</i>	165
4.4.2	<i>Bilan de la démarche EA4UP</i>	168
5.	Démarche EA4UP et mesure d'alignement fonctionnel de type information mutuelle moyenne	169
5.1	Principes	169
5.1.1	<i>Définition de l'alignement fonctionnel</i>	169
5.1.2	<i>Axiomatisation de l'alignement fonctionnel</i>	173
5.1.3	<i>Choix d'une mesure de l'alignement fonctionnel</i>	174
5.2	Démarche EA4UP et théorie de l'information	176
5.2.1	<i>Analyse fonctionnelle dynamique et transmission d'information</i>	176
5.2.2	<i>PLU fonctionnel et contrainte de transmission</i>	176
5.3	Alignement fonctionnel et système de communication	177
5.3.1	<i>Définition du système de communication</i>	178
5.3.2	<i>Codage d'information et approche MDA</i>	179
5.3.3	<i>Décodage d'information et approche MDA</i>	181
5.4	PLU fonctionnel et canal de communication	183
5.4.1	<i>Canal de communication et approche MDE</i>	183
5.4.2	<i>Niveaux du PLU et loi de transition</i>	183
5.5	Mesure d'alignement fonctionnel et information mutuelle moyenne	188
5.5.1	<i>Choix de l'information mutuelle moyenne</i>	188
5.5.2	<i>Formule de l'information mutuelle moyenne pour l'alignement fonctionnel</i>	189
5.6	Conclusion	194
6.	Conclusion	195
7.	Perspectives	197
8.	Références bibliographiques	199
9.	Liste des publications personnelles et des brevets	205
10.	Acronymes	207
11.	Table des règles d'urbanisme	209
12.	Table des Figures	211
13.	Table des Tableaux	215

1. Introduction

1.1 Contexte

La réussite du management d'une entreprise repose classiquement sur cinq points principaux : fixer des objectifs, organiser le travail, motiver et communiquer, mesurer la performance, former les salariés [Drucker, 1983].

La fixation des objectifs sous-entend une stratégie de l'entreprise dont sont dérivés ces objectifs. La stratégie permet principalement la prise en compte de l'évolution d'un marché. Dans le cas d'un opérateur de télécommunications, elle doit aussi anticiper les ruptures technologiques. L'entreprise doit donc s'adapter à l'évolution de son environnement et à l'évolution de sa propre stratégie. Il est important de souligner l'accélération actuelle de la stratégie de croissance des entreprises. Cette stratégie de croissance est matérialisée par une création de moyens de production ou de financement nouveaux, par une fusion avec d'autres entreprises, etc.

L'organisation de l'entreprise doit répondre à sa stratégie. Parmi les facteurs de réussite cités par Peter Drucker, le travail organisé nécessite des standards ou normes. Conséquence du taylorisme, le travail dans l'entreprise est défini méthodologiquement par des modes opératoires qui sont des successions de tâches. Ces modes opératoires permettent d'atteindre les objectifs de production de l'entreprise. Ils nécessitent une structure organisationnelle, c'est-à-dire des unités organisationnelles et un réseau qui les relie. Un élément de la structure organisationnelle contribue alors à une ou plusieurs tâches de ces modes opératoires. Cette contribution est en général affectée à un rôle. Par exemple, la vente d'un produit est affectée à un rôle de vendeur. Lors de la vente d'un produit, l'unité organisationnelle du vendeur est reliée avec l'unité organisationnelle de facturation responsable de la facturation du produit.

En plus de son organisation qui la caractérise, l'entreprise est identifiée depuis une quinzaine d'années par les processus qui décrivent son cœur de métier et non plus à partir de la décomposition du travail en tâches créée par Adam Smith en 1776. Il s'agit de l'approche de réingénierie où le processus est défini comme une série d'activités qui, ensemble, produisent une valeur pour un client [Hammer, 1993]. La description d'un processus explique comment les ressources de l'entreprise (compétences, machines, main d'œuvre) sont organisées pour produire cette valeur. La réingénierie, qui est née en Amérique du Nord, permet de s'adapter à l'accélération de l'évolution des besoins du client, à la concurrence de plus en plus forte et au changement dans l'entreprise et en dehors de l'entreprise. Concernant les compétences et la main d'œuvre, les rôles utiles à l'entreprise sont joués par ses ressources humaines. Elles permettent à l'entreprise de définir son organisation et ses objectifs. C'est donc à l'entreprise de faire correspondre au mieux ses ressources humaines avec les rôles qui contribuent aux processus définissant son cœur de métier [Crozier, 1994]. La responsabilité de chaque rôle définie sur chaque processus permet en effet de dimensionner correctement les ressources nécessaires à chaque processus. L'optimisation du dimensionnement des processus revient à coordonner les activités dédiées à chaque rôle de l'entreprise de manière à minimiser le coût des ressources nécessaires [Burlton, 2001].

Le travail organisé nécessite aussi la mise à disposition d'informations sur la production dans l'Entreprise. La mise à disposition de ces informations est décrite dans un Système d'Information (SI) de l'entreprise. La stratégie de l'entreprise a des conséquences sur le SI. L'accélération de la stratégie de croissance de l'entreprise, l'évolution de son organisation et des processus auxquels elle contribue sont source de complexité pour l'entreprise [Morin, 1990]. Cette complexité nécessite des modèles de compréhension et des modèles d'intervention. Les modèles représentent les interactions des éléments de l'entreprise nécessaires à la production. La complexité d'un Système d'Information est par exemple liée aux interactions des systèmes qui le constituent. Le système est une unité complexe dépendante de son objet, mais aussi de son organisation et de ses associations avec son environnement. Par analogie avec les types de modèles proposés par Edgar Morin, l'objectif de l'Architecture d'Entreprise (EA : Enterprise Architecture) d'un SI [Zachman, 1987] serait de proposer :

- un modèle de compréhension en décrivant l'existant du SI, par exemple un système de commande en agence commerciale, un système de commande par l'intermédiaire d'un portail, un système de facturation et le lien existant entre eux,
- un modèle d'intervention en définissant une cible pour le SI, par exemple un nouveau système de commande unifiée et son lien avec le système de facturation existant.

Cette description de l'existant, ou cartographie, a une autre utilité lorsqu'elle cible les processus métier de l'entreprise. La cartographie permet en effet de reconfigurer les processus et donc de permettre une durabilité des grandes fonctions de l'entreprise (fonction stratégique, fonction opérationnelle, fonction logistique). La durabilité de ces fonctions est définie par l'aptitude de l'entreprise à évoluer au gré des transformations du marché, de l'évolution des technologies ou des fusions d'entreprise. Les technologies de l'information sont un catalyseur important de cette durabilité. Il est intéressant de noter que la mise en œuvre de ces technologies de l'information dans la réingénierie est de nature inductive et non déductive [Hammer, 1993]. Ceci signifie que la détection d'une solution permet de résoudre des problèmes postérieurs à la solution. C'est un fondement de l'EA au travers de la définition d'un SI cible. L'objectif de l'EA est en effet, à partir d'une cartographie du SI de l'entreprise, de définir un SI cible permettant de supporter les évolutions du marché avant que ces évolutions ne soient définies de façon détaillée. La stratégie prise en compte en entrée de la définition du SI cible a pour objet de décrire de façon macroscopique certaines de ces évolutions. C'est donc à partir de cette stratégie de l'entreprise que la cible de l'EA est définie.

La complexité des systèmes induit des comportements inattendus. L'EA doit favoriser la maîtrise des comportements inattendus des systèmes d'un même SI d'une entreprise. C'est par exemple le cas lorsqu'une même donnée décrivant le client d'une entreprise est modifiée dans des systèmes différents de façon non coordonnée. L'objectif de l'EA est alors de diminuer les comportements incertains en proposant un modèle architectural de référence du SI. Dans l'exemple proposé, un des systèmes sera responsable de la mise à jour de la donnée décrivant un client et l'autre système interagira avec lui pour récupérer la donnée. Cette diminution nécessite d'axer l'EA sur la stabilité dynamique du SI et des systèmes de ce SI. Les comportements dynamiques d'un SI, d'un système ou de l'entreprise avec sa structure organisationnelle sont analogues. Dans ces trois cas illustrant l'approche systémique, il s'agit d'ensemble d'unités et d'interactions entre ces unités [Rosnay, 1975]. L'approche systémique englobe les éléments du système au sens large ainsi que leurs interactions. La vision dynamique intègre le temps afin de souligner le comportement du système selon des scénarios d'usage du système. L'objectif de stabilité dynamique permet l'évolutivité du système et sa durabilité.

Les tâches des modes opératoires de l'entreprise sont automatisées ou non dans son SI. L'alignement du SI sur l'organisation de l'entreprise devient un indicateur pertinent de la durabilité du SI par rapport à celle de l'entreprise. La durabilité de l'entreprise est mise en valeur par l'intégration de sa stratégie dans les processus de l'entreprise qu'elle utilise pour produire de la valeur. Un SI durable est mis en valeur par l'intégration de cette stratégie dans les systèmes du SI de l'entreprise qu'elle utilise pour produire de la valeur. Un des indicateurs de la durabilité du SI est l'alignement de chaque système du SI avec l'EA représentant les impacts de la stratégie sur le SI. Dans l'exemple du SI cible comportant un système de commande et un système de facturation, la création d'un système de commande spécifique à un type de produit tel qu'un téléphone du réseau mobile pour un opérateur de télécommunications est source de désalignement avec le SI cible. L'intégration de la commande du téléphone du réseau mobile au système de commande unifié aurait par contre induit un alignement parfait avec le SI cible.

1.2 Contribution

Comme l'indique Edgar Morin dans [Morin, 1990], "Le refus de la complexité génère le désordre et la confusion". La complexité due à l'évolution de l'entreprise rend nécessaire la durabilité de son SI. La contribution de cette thèse est définie comme une offre d'assistance aux architectes dans leur volonté de diminuer le désordre et la confusion lors de la conception d'un système ou d'un SI :

- *Assister la modélisation de l'EA fonctionnelle d'un SI, c'est-à-dire la modélisation fonctionnelle du SI, en s'appuyant sur une vue dynamique de ce SI.* L'objectif serait, par exemple, de faciliter la conception de la vue fonctionnelle d'un SI de commande à partir de scénarios fonctionnels liés à la commande tels que la création d'une commande d'un téléphone du réseau mobile en agence, la création d'une commande d'un service télécom de messagerie sur un portail, etc.
- *Assister le développement d'un système en s'appuyant sur l'EA, sur l'ingénierie des modèles afin d'intégrer le modèle de l'EA fonctionnelle du SI et sur une vue dynamique du système.* Dans l'exemple du SI de commande, l'objectif serait de faciliter le développement d'un système de commande sur un portail en intégrant l'EA fonctionnelle du SI et en étudiant les différents usages du système par l'intermédiaire des différents scénarios de commande sur un portail tels que la création d'une commande, la modification d'une commande, etc.
- *Assister l'architecte fonctionnel d'un système en proposant une mesure de l'alignement de son système avec l'EA fonctionnelle du SI qui s'appuie sur les principes de la théorie de l'information et sur une vue dynamique du système.* La réutilisation d'éléments permettant de diminuer les coûts, l'objectif est d'évaluer l'effort en termes de ressources pour développer ce système. Toujours dans le cadre du SI de commande et du système de commande sur un portail, l'objectif est de donner au développeur du système une indication de l'écart entre le modèle fonctionnel du système de commande sur un portail qu'il a conçu et l'EA fonctionnelle du SI.

1.3 Plan du document

Suite à un état de l'art décrit dans le chapitre 2, la contribution de la thèse est détaillée dans les trois chapitres suivants. Le chapitre 3 présente la démarche DA4EA (Dynamic Approach for Enterprise Architecture) permettant de modéliser l'EA fonctionnelle d'un SI. La démarche DA4EA est illustrée et évaluée avec la vue fonctionnelle du SI supportant le processus métier d'usage d'un service télécom.

Le chapitre 4 présente la démarche EA4UP (Enterprise Architecture for Unified Process) de développement d'un système du SI ainsi que son évaluation. L'utilisation de l'ingénierie des modèles dans les activités de conception d'architecture d'EA4UP est décrite dans ce chapitre ainsi que son illustration avec un cas d'utilisation de service télécom. Un bilan d'EA4UP suite à son prototypage par des projets de développement de services télécom complète la description de la démarche.

Le chapitre 5 présente une mesure d'alignement entre l'architecture d'un système du SI et l'EA fonctionnelle du SI. Les concepts d'EA4UP sont rapprochés de concepts de la théorie de l'information. Ce rapprochement permet de présenter l'information mutuelle moyenne comme mesure d'alignement. Une illustration de ces mesures d'alignement est présentée avec quelques cas d'utilisation de services télécom.

Une conclusion et des perspectives à court, moyen et long termes de la contribution de la thèse sont proposées respectivement dans le chapitre 6 et dans le chapitre 7.

2. État de l'art

2.1 Architecture d'entreprise

2.1.1 Durabilité d'un SI et réutilisabilité

Le Système d'Information (SI) représente l'ensemble des ressources humaines, des ressources matérielles et des ressources logicielles interconnectées afin de collecter, stocker, traiter et communiquer les informations. Le système informatique est la partie du SI réduite aux ressources logicielles [Brien, 1997]. Le SI coordonne les activités de l'organisation de l'entreprise grâce à l'information et permet ainsi d'atteindre les objectifs de l'entreprise. L'évolution des objectifs de l'entreprise est donc un facteur central de l'évolution d'un SI. Un objectif de vente de produits au travers d'un portail sur Internet provoque par exemple une évolution du SI d'une entreprise où la vente ne se fait qu'en agence de distribution.

Cette évolution implique :

- des impacts fonctionnels, si l'entreprise considère que cette fonction de vente est différente d'une fonction de vente classique. Cela peut être le cas avec une notion de caddie virtuel permettant de modifier la commande en temps réel,
- des impacts techniques avec les technologies liées à l'Internet,
- des impacts applicatifs avec l'extension ou la création d'une nouvelle application réalisant cette vente par correspondance déployée sur Internet.

Le problème de la réutilisabilité de l'existant est alors de gérer au mieux les évolutions de l'entreprise au travers de son SI. La réutilisabilité d'éléments du SI est donc un facteur de sa durabilité (cf. §1.1).

L'architecture orientée objet, l'architecture orientée composants et l'architecture orientée services sont au cœur de la conception des systèmes d'un SI [Wang, 2004]. Le service est fondé sur plusieurs composants, par exemple un service de commande peut s'appuyer sur un composant de gestion des commandes, un composant de gestion du client et un composant de gestion du catalogue. Le composant est implémenté à partir de technologies orientées objet telles qu'UML, Java, etc. Un composant est caractérisé par le fait qu'il peut interagir avec d'autres composants. Dans l'exemple, le composant de gestion de la commande interagit avec le composant de gestion du client afin d'identifier le client de la commande. La conception à partir de composants permet la réutilisabilité de composants existants dans l'entreprise, et donc favorise la diminution des coûts de développement. L'architecture orientée objet permet de modéliser l'architecture orientée composants.

L'intérêt de l'architecture orientée services (SOA) est de permettre l'implémentation d'enchaînements de tâches réalisant un processus métier. L'architecture orientée services fournit une méthodologie permettant d'organiser et d'utiliser des ressources du système informatique de manière distribuée [Schulte, 1996]. L'architecture SOA peut ainsi favoriser la réutilisation des services offerts dans le SI de l'entreprise lors du développement de tous nouveaux systèmes. La difficulté est néanmoins de bien définir ces services afin de les réutiliser de la manière la plus pertinente, en particulier par rapport aux fonctions que réalisent ces services.

La modélisation des processus métier peut bénéficier d'une approche orientée services [Cauvet, 2008]. Les services orchestrés pour représenter un processus métier peuvent être des services métier définis à l'aide d'un vocabulaire partagé dans l'entreprise et modélisant une partie d'un processus métier. Cette partie d'un processus reste spécifique aux objectifs de l'entreprise et aux ressources utiles à sa réalisation. Le lien avec la stratégie de l'entreprise favorise la conception de l'architecture du processus métier par rapport à l'enchaînement des tâches liées aux interactions avec le client. La réutilisation des services est ainsi facilitée.

La durabilité d'un SI dans le cas d'une architecture de ses systèmes orientée composants repose sur l'architecture de la vue métier de l'entreprise et sur les modèles d'analyse de la vue fonctionnelle du SI [Fingar, 2000]. Un référentiel de composants conçus à partir de ces éléments doit favoriser la réutilisation de ceux-ci lors de leur assemblage dans un système. La stratégie de l'entreprise reste le point de départ du développement des systèmes car elle est la source de la spécification du référentiel de processus et du référentiel de modèles d'analyse fonctionnels.

La réutilisation peut aussi s'appuyer sur des patrons définissant des problèmes récurrents et la solution associée. C'est le cas des patrons définis pour les SI produits qui recouvrent l'organisation de l'entreprise et le système informatique participant au développement d'un produit de l'entreprise [Cauvet, 2000]. L'architecture fonctionnelle cible d'un SI peut être considérée comme un patron fonctionnel favorisant la réutilisation de composants applicatifs. En effet, la conception des patrons repose sur une démarche inductive telle que recommandée pour la réingénierie (cf. §1.1). De plus, l'utilisation de modèles de composants assimilables à des patrons et favorables à la réutilisation est préconisée pour le développement de grand systèmes [Ambler, 1998]. Elle permet de simplifier les développements orientés objet de tels systèmes.

2.1.2 Généralités sur l'EA

L'évolution des SI s'accélère du fait de l'informatisation grandissante des entreprises et des nouveautés technologiques. La difficulté pour une entreprise est alors de conduire l'évolution du SI au lieu de la subir. C'est le principe fondateur de l'architecture d'entreprise ou EA (Enterprise Architecture). L'EA offre une représentation des connaissances d'une entreprise et la manière dont elles sont supportées [Zachman, 1987]. Cette représentation recouvre :

- les descriptions de l'organisation de l'entreprise et de ses objectifs,
- les différentes modélisations de ses processus métier, de son SI et de son système informatique.

L'EA offre plusieurs points de vue du SI. Ces points de vue sont aussi bien celui des décideurs axé sur la stratégie de l'entreprise, que celui des développeurs comprenant l'architecture des systèmes du SI. Pour ce point de vue de l'architecture des systèmes du SI, l'objectif est de résoudre les exigences fonctionnelles ou non fonctionnelles parfois contradictoires dans un SI et d'optimiser les échanges entre ces systèmes. C'est la problématique des très gros systèmes (ULS system : Ultra-Large-Scale system) appliquée à un SI considéré comme un système de systèmes et à ses règles de conception [Baldwin, 2000]. La solution proposée par l'EA est de découper ce point de vue des systèmes en vues, ou visions, complémentaires telles qu'une vue métier ou une vue technique du SI. Ces vues deviennent la référence pour tous les systèmes du SI.

L'EA est fondée de façon macroscopique sur quatre processus [Urba-EA, 2006] :

- la description des ressources de l'entreprise (humaines ou logicielles) interagissant avec le SI de l'entreprise. Cette description de ressources, ou cartographie, permet de gérer l'évolution du SI en répertoriant l'existant. Parmi les ressources logicielles, la donnée est au cœur du système d'information. Elle représente l'information traitée par le SI. Après le recensement des applications, un des premiers objectifs de la cartographie est de définir les données référentielles de chaque application. L'objectif est de définir l'application responsable de chaque donnée du SI [Urba-SI, 2003] ;
- la définition d'une EA cible par rapport aux axes stratégiques de l'entreprise. Lorsque cette cible est atteinte, le SI est dit durable. Les éléments du SI étant en phase avec la stratégie de l'entreprise, ils ont en effet une durée de vie plus longue et deviennent réutilisables pour de nouveaux développements de systèmes ;
- la migration du SI existant vers un SI conforme à l'EA cible. Cette migration est en général spécifiée dans un schéma directeur de l'entreprise qui donne une vision stratégique de cette entreprise à 3 ou 4 ans ;
- la mise en œuvre de la cible par les projets de développement de système. Il s'agit pour ces projets d'intégrer l'EA cible lors de la conception de l'architecture d'un système.

Les rôles contribuant à ces processus sont :

- la maîtrise d'ouvrage (MOA) du projet d'EA, soit le client du projet de conception de l'EA,
- la MOA du projet de développement d'un système, soit le client de ce projet,
- l'architecte d'entreprise qui conçoit les différents points de vue de l'EA,
- le cartographe du SI qui cartographie les systèmes existants du SI,
- l'architecte du système à développer dans le SI.

État de l'art

Les contributions de chaque rôle dans les quatre processus liés à l'EA sont représentées dans le diagramme d'activités UML (cf. §2.1.5) de la **Figure 1**.

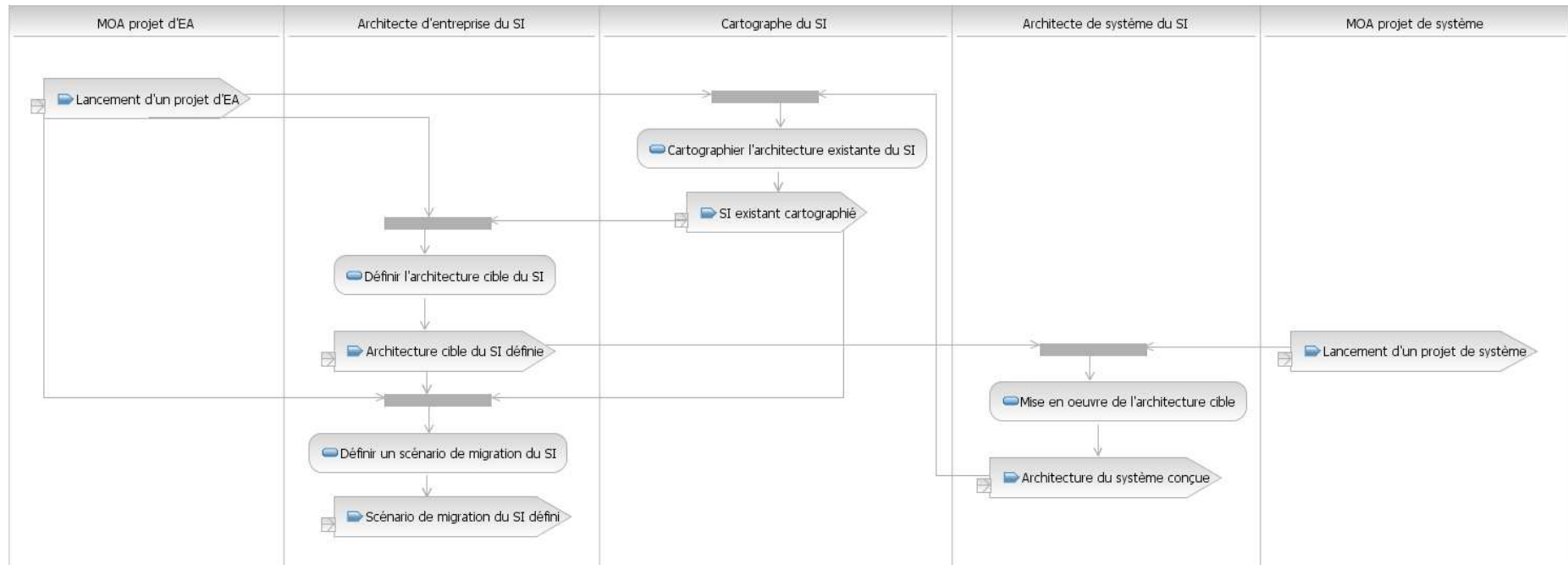


Figure 1 : Processus liés à l'EA et rôles contributeurs

Les événements déclencheurs des ces processus lié à l'EA sont définis de la façon suivante :

- le lancement d'un projet d'EA qui répond à la définition par l'entreprise d'une stratégie,
- le lancement d'un projet de système qui répond à la définition du besoin d'un utilisateur de ce système.







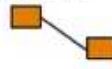

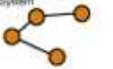
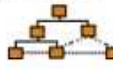


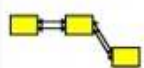
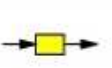
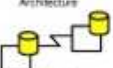
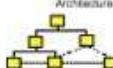


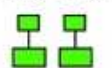
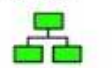

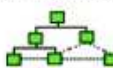








La durabilité d'un SI est favorisée par les apports de l'EA appliquée à un SI :

- la diminution des coûts de développement et de maintenance due à la réutilisation d'éléments du SI satisfaisant les préconisations de l'EA,
- la facilitation de la migration technique d'un SI (changement d'OS, etc.) ou de la migration fonctionnelle d'un SI (passage d'un client du marché entreprise ou du marché grand public à un client du réseau fixe ou du réseau mobile pour un opérateur de télécommunication, etc.) à partir de la cartographie complète du SI existant et de la définition de l'EA cible,
- l'homogénéité des solutions technologiques utilisées et leur cohérence avec les exigences techniques spécifiques à l'entreprise (nombre de client, taille et configuration du réseau de l'entreprise, etc.),
- l'assistance efficace à la fusion d'entreprise à partir du partage du point de vue métier des entreprises fusionnées.

2.1.3 Différents cadres de l'EA

Pour améliorer la durabilité du SI, les architectes d'entreprise disposent de différents cadres. Ces cadres définissent une EA reposant sur différents points de vue de l'entreprise (point de vue métier, point de vue organisationnel, point de vue des systèmes, point de vue technique, etc.). Ces points de vue permettent d'évaluer les impacts des différents axes stratégiques définis par l'entreprise.

La plupart de ces cadres sont dérivés du cadre de référence de Zachman représenté dans la **Figure 2** [Zachman, 1987].

A FRAMEWORK FOR ENTERPRISE ARCHITECTURE™							
	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	PEOPLE <i>Who</i>	TIME <i>When</i>	MOTIVATION <i>Why</i>	
SCOPE (CONTEXT)	List of Things Important to the Business 	List of Processes the Business Performs 	List of Locations in which the Business Operates 	List of Organizations Important to the Business 	List of Events/Cycles Significant to the Business 	List of Business Goals/Strategies 	SCOPE (CONTEXT)
<i>Planner</i>	ENTITY = Class of Business Thing	Process = Class of Business Process	Node = Major Business Location	People = Major Organization Unit	Time = Major Business Event/Cycle	Ends/Mean = Major Business Goal/Strategy	<i>Strategist</i>
BUSINESS MODEL (CONCEPTS)	e.g. Semantic Model 	e.g. Business Process Model 	e.g. Business Logistics System 	e.g. Work Flow Model 	e.g. Master Schedule 	e.g. Business Plan 	BUSINESS MODEL (CONCEPTS)
<i>Owner</i>	Ent = Business Entity Reln = Business Relationship	Proc = Business Process IO = Business Resources	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	End = Business Objective Means = Business Strategy	<i>Executive Leaders</i>
SYSTEM MODEL (LOGIC)	e.g. Logical Data Model 	e.g. Application Architecture 	e.g. Distributed System Architecture 	e.g. Human Interface Architecture 	e.g. Processing Structure 	e.g. Business Rule Model 	SYSTEM MODEL (LOGIC)
<i>Designer</i>	Ent = Data Entity Reln = Data Relationship	Proc = Application Function IO = User Views	Node = I/O Function (Processor, Storage, etc.) Link = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	End = Structural Assertion Means = Action Assertion	<i>Architects</i>
TECHNOLOGY MODEL (PHYSICS)	e.g. Physical Data Model 	e.g. System Design 	e.g. Technology Architecture 	e.g. Presentation Architecture 	e.g. Control Structure 	e.g. Rule Design 	TECHNOLOGY MODEL (PHYSICS)
<i>Builder</i>	Ent = Segment/Table/etc. Reln = Pointers/Keys/etc.	Proc = Computer Function IO = Data Elements/Sets	Node = Hardware/Systems/Software Link = Line Specifications	People = User Work = Screen Format	Time = Execute Cycle = Component Cycle	End = Condition Means = Action	<i>Engineers</i>
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	e.g. Data Definition 	e.g. Program 	e.g. Network Architecture 	e.g. Security Architecture 	e.g. Timing Definition 	e.g. Rule Specification 	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
<i>Sub-Constructor</i>	Ent = Field Reln = Address	Proc = Language Statement IO = Control Block	Node = Address Link = Protocol	People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	End = Subcondition Means = Step	<i>Implementers</i>
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

© 1986 - 2008 John A. Zachman, Zachman International

See www.ZachmanInternational.com for the latest Zachman Framework graphic.

Figure 2 : Cadre de référence de Zachman

Les cadres de l'EA sont très, voire trop, nombreux [Schekkerman, 2006]. Certains d'entre eux sont devenus des références pour l'EA :

- le TOGAF (The Open Group Architecture Framework [TOGAF, web],
- le DoDAF (Department of Defense Architecture Framework) [DoDAF, web] et ses dérivés de l'OTAN ou du Royaume Uni,
- l'ITIL (Information Technology Infrastructure Library) [ITIL, web],
- le RM-ODP (Reference Model for Open Distributed Processing) [RM-ODP, web].

Le TOGAF, créé par un consortium industriel (Open Group fondé par IBM, Sun Microsystems, Hitachi, Hewlett-Packard, Fujitsu), propose une méthode ADM (Architecture Development Method) et un ensemble d'outils de conception de l'architecture d'entreprise en complément d'un environnement d'architecture. La méthode ADM de conception de l'EA est fondée sur la gestion des exigences au niveau de l'entreprise. Ces exigences sont déduites des objectifs et des orientations métier de l'entreprise. L'architecture de chaque point de vue conçue à partir de ces exigences cible le métier de l'entreprise et le SI. Le TOGAF cible de plus la définition des technologies qui répondent le mieux aux exigences de l'entreprise. Ces technologies composent alors l'infrastructure du SI

Le DoDAF est un cadre de l'EA du département de la défense américain issu du FEA (Federal Enterprise Architecture Framework) proposé en 1999 pour les agences ou les départements du gouvernement fédéral. Il est centré autour de la préconisation de l'architecture SOA et des web services. De ce fait, il permet d'assurer l'interopérabilité des grands systèmes du SI de la défense américaine. Il propose :

- un point de vue organisationnel avec les échanges entre acteurs de la défense,
- un point de vue système avec les échanges entre systèmes,
- un point de vue technique avec les technologies préconisées.

Le cadre ITIL est une bibliothèque de bonnes pratiques liées à des exigences de performance et de qualité des systèmes développés. Ces bonnes pratiques ciblent le service après-vente des systèmes en s'assurant des performances et de la qualité perçues, d'une part, par le client du système, et de l'autre, par l'entreprise titulaire du SI contenant le système. Dans ce dernier cas, les préconisations sont aussi bien financières que liées à la continuité de service ou à la disponibilité des systèmes du SI.

RM-ODP est un cadre recommandé par l'ITU (International Telecommunication Union). Ce modèle de référence est particulièrement intéressant du fait qu'il est proche du processus de développement d'un système. RM-ODP fournit en effet un point de vue de l'information supportée par un système, un point de vue logiciel, un point de vue de l'ingénierie et un point de vue technique. De plus, ces points de vue sont liés. Le lien entre points de vue préfigure une modélisation des concepts de chacun des points de vue.

2.1.4 Un cadre particulier de l'EA : l'urbanisme

Ces différents cadres de l'EA sont volontairement peu précis quant à la modélisation de leurs différents points de vue. Cette volonté permet d'afficher une plus grande souplesse vis-à-vis des contraintes des multiples techniques et outils de modélisation de chaque entreprise [Salinesi, 2008]. De plus, l'absence d'un aspect hiérarchique de cette modélisation pour le point de vue système est pénalisante lorsque le métier d'une entreprise est spécifié par des dizaines de processus métier et lorsque son système informatique comprend plusieurs centaines d'applications. La découpe par niveau est en effet très utile pour rechercher une fonction supportant un processus du SI, par exemple la fonction de création d'une commande d'un produit, et obtenir la liste des applications du système informatique qui la réalise. De même lors du développement d'un système, une modélisation de type hiérarchique permet aux architectes de détecter plus efficacement des éléments du SI afin de les réutiliser.

Par rapport à ce manque, un cadre intéressant de l'EA est celui de l'urbanisme [Sassoon, 1998]. Le cadre de l'urbanisme a été développé en France par des entreprises de taille significative au début des années 90. Il offre une partie du point de vue métier, avec la description des processus du cœur de métier de l'entreprise, et le point de vue système décrits dans le cadre de référence de Zachman (cf. **Figure 2**). Plus précisément, l'urbanisme cible à l'intérieur de ces points de vue :

- le quoi, soient les données,
- le comment, soient les fonctions,
- le où, soit le réseau,
- et une partie du qui, soient les rôles.

La spécificité de ce cadre est l'enrichissement de ces points de vue par une approche née de l'urbanisme des villes. Dans l'approche systémique décrite dans le §1.1, le SI et la ville sont en effet considérées comme des systèmes, puisque tous deux sont constitués d'unités et d'interactions entre unités [Rosnay, 1975]. Les unités et leurs interactions sont décrites dans le PLU (Plan Local d'Urbanisme), anciennement le POS (Plan d'Occupation des Sols), pour la ville et de façon plus générale pour la commune [PLU, 2008]. Le PLU d'une commune comporte un projet d'aménagement et de développement durable qui définit les orientations générales d'aménagement et d'urbanisme retenues pour l'ensemble de la commune. Il doit répondre en particulier à des besoins d'aménagement de l'espace.

Du fait de l'analogie entre ville et SI [Contini, 2002], l'aménagement de l'espace équivaut à localiser les fonctions du SI dans un îlot fonctionnel unique, lui-même inclus dans un quartier fonctionnel unique contenu, à son tour, dans une zone fonctionnelle unique (cf.

Figure 3).

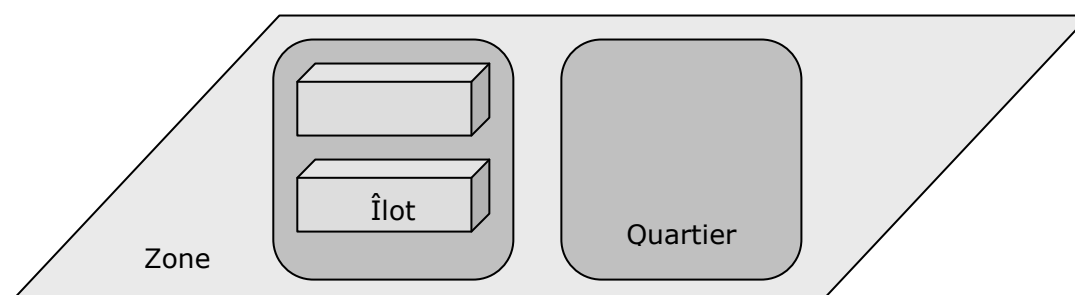


Figure 3 : Zone, quartier et îlot du PLU fonctionnel

Plus précisément, le PLU permet de découper l'architecture fonctionnelle du SI par niveau :

- Le niveau îlot est le plus bas niveau de découpe du SI. Ce niveau est celui où chaque élément peut être conçu indépendamment des autres. Dans le cas de la vue fonctionnelle à laquelle le PLU est appliqué, un îlot fonctionnel de *Gestion des contacts avec le client* et un îlot fonctionnel de *Gestion des commandes* pourraient être conçus pour le SI d'une entreprise supportant un processus métier de vente ;
- Le niveau quartier est tel que chaque quartier est un regroupement homogène d'îlots quant à la nature de l'information traitée. Dans le cas de l'exemple du SI de vente, les îlots fonctionnels de *Gestion des contacts avec le client* et de *Gestion des commandes* pourraient être regroupés dans un quartier fonctionnel de *Gestion de la relation commerciale* ;
- Le niveau zone est le plus haut niveau de l'organisation de la vue fonctionnelle du SI. Toujours dans le SI de vente, la zone fonctionnelle *Distribution* pourrait par exemple être conçue, et inclure le quartier fonctionnel de *Gestion de la relation commerciale*.

Un complément important de la définition du concept de quartier pour la démarche DA4EA (cf. §1.2) est de considérer un quartier comme un ensemble d'îlots fonctionnels ayant une cohésion forte et un couplage faible [Larman, 2005].

Chaque îlot fonctionnel du PLU est découpé dans la **Figure 4** en parcelles qui correspondent à des fonctions élémentaires. L'îlot fonctionnel de *Gestion des commandes* comprendrait par exemple les parcelles fonctionnelles de *Création d'une commande* et de *Consultation d'une commande*.

De plus, des voies fonctionnelles peuvent exister entre îlots. Ces voies permettent des interactions entre les îlots fonctionnels. L'existence d'une voie entre les îlots fonctionnels de *Gestion des contacts avec le client* et de *Gestion des commandes* est probable dans un SI dédié à la vente. Cette voie représenterait la dépendance fonctionnelle existante entre la prise de contact avec un client et la possibilité que ce client ait de passer une commande.

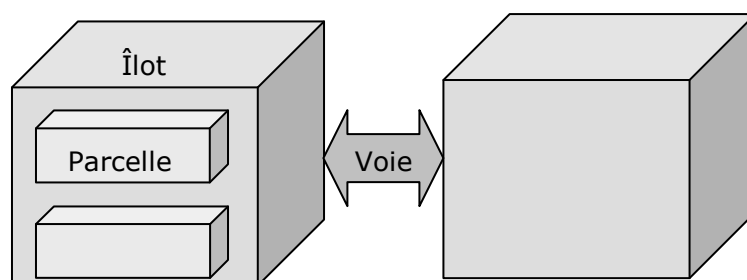


Figure 4 : Îlot, parcelle et voie du PLU fonctionnel

2.1.5 EA et modélisation

Le SI d'une entreprise est un ensemble complexe. Il est en effet composé de plusieurs centaines de systèmes pour de grandes entreprises et de multiples interactions entre ces systèmes. Le processus de cartographie des systèmes du SI (cf. §2.1.2) permet de résoudre cette complexité par la modélisation de l'EA du SI. La modélisation d'après Grady Booch est en effet une représentation simplifiée de la réalité permettant de comprendre un système en cours de développement.

Les modèles sont constitués ici de diagrammes UML (Unified Modeling Language) [Booch, 1999] [UML, web] ou d'éléments textuels. Les diagrammes UML les plus utilisés sont les diagrammes d'activité, les diagrammes de classes et les diagrammes de séquence. Dans le cas d'un processus d'usage d'un service télécom de messagerie, incluant une activité de filtrage des messages permettant de renforcer la sécurité lors de la réception d'un message, les utilisations de ces diagrammes sont par exemple :

- le diagramme d'activités, illustré par la **Figure 5**, dont les concepts représentant un processus sont
 - o un couloir de responsabilité avec le rôle responsable de l'activité (*Utilisateur du service de messagerie*)
 - o un événement déclenchant le processus (*Demande de messages reçus*),
 - o une activité (*Réception d'un message*),
 - o une transition (*Filtrage appliqué au message succède à Réception d'un message*),
 - o un point de synchronisation entre activités (après l'évènement *Demande de messages reçus* et la branche conditionnelle *Autre message*),
 - o un point de décision (avant les branches conditionnelles *Autre message* et *Pas d'autre message*).

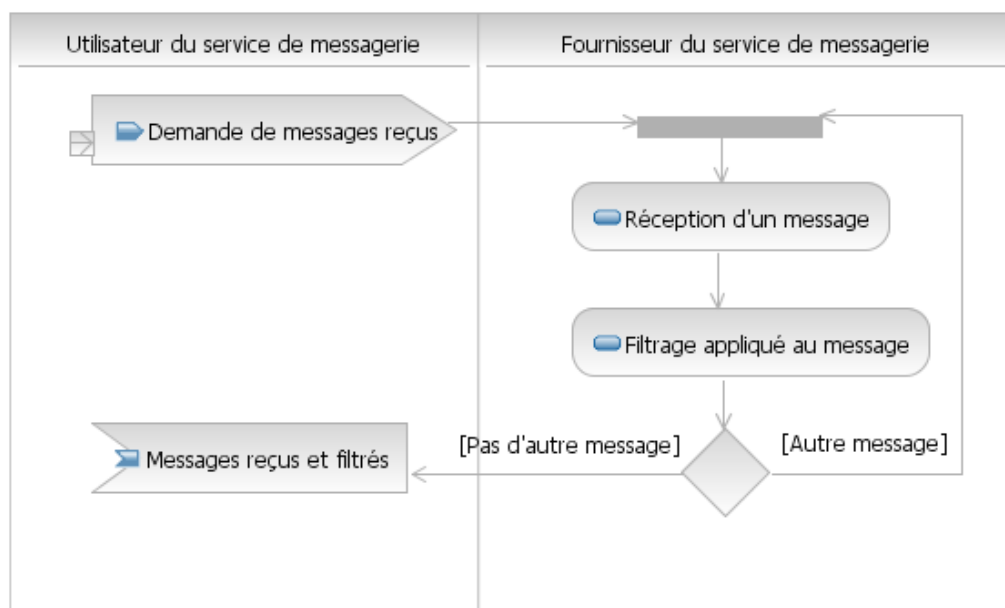


Figure 5 : Illustration d'un diagramme d'activités UML

- le diagramme de classes, illustré par la **Figure 6**, dont les concepts sont :
 - o une classe (*Sécurité de la messagerie – Gérer les règles de filtrage*),
 - o un attribut ou une opération de la classe (l'opération *Créer les règles de filtrage de message*),
 - o une relation de type association ou dépendance (la dépendance *dépend de* entre les classes *Sécurité de la messagerie – Gérer les règles de filtrage* et *Sécurité de la messagerie – Filtrage de message*).



Figure 6 : Illustration d'un diagramme de classes UML

- le diagramme de séquence, illustré par la **Figure 7**, dont les concepts représentant un scénario sont :
 - o un scénario ou instance de cas d'utilisation (l'instance du cas d'utilisation *Sécurité renforcée lors de la réception d'un message électronique*),
 - o un objet ou une instance de classe (l'instance de la classe *Sécurité de la messagerie – Gérer les règles de filtrage*),
 - o une interaction entre instances de cas d'utilisation et de classe, ou entre instances de classes, spécifiée par un message (le message *Créer les règles de filtrage de message* entre les instances des classes *Sécurité de la messagerie – Gérer les règles de filtrage* et *Sécurité de la messagerie – Filtrage de message*).

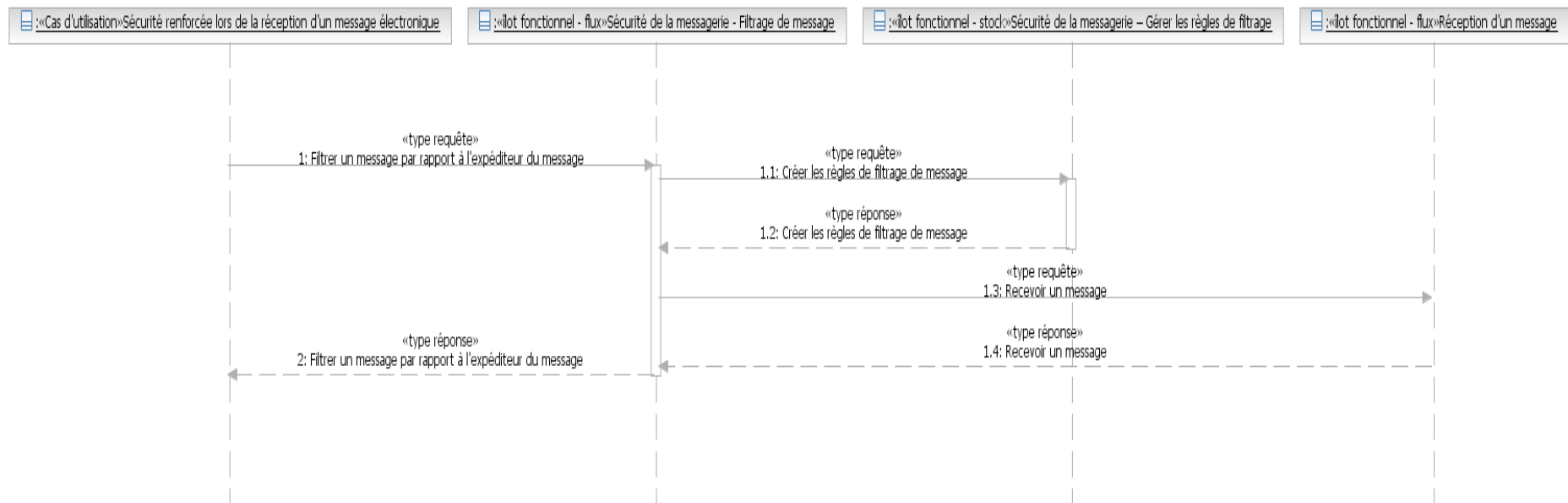


Figure 7 : Illustration d'un diagramme de séquence UML

Chaque concept définissant un diagramme UML peut être typé par un stéréotype représenté entre guillemets. Le stéréotype peut par exemple préciser un concept du diagramme par son rôle dans l'EA du SI. Par exemple, la classe UML *Sécurité de la messagerie – Gérer les règles de filtrage* représente un îlot fonctionnel de type stock défini dans l'EA du SI des services télécom (cf. **Figure 8**).

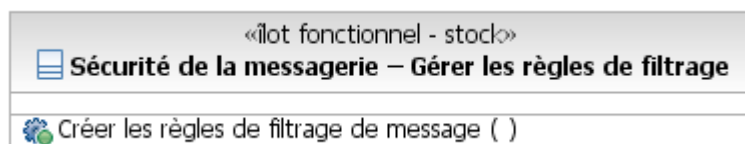


Figure 8 : Illustration d'un stéréotype d'une classe UML

La traduction d'une relation de dépendance d'un diagramme de classes UML par une interaction d'un diagramme de séquence est précisée dans UML par la Sémantique UML – Approche dynamique et diagramme de séquence rappelée ci-après. Cette règle définit une traduction d'un diagramme de séquence UML décrivant un scénario d'usage d'un système.

Sémantique UML – Approche dynamique et diagramme de séquence

Dans un diagramme de séquence UML :

- une succession d'interactions ayant pour origine commune la ligne de vie d'un scénario, ou la ligne de vie d'une instance d'une classe représente un ordonnancement temporel de ces interactions,
- une interaction entre des instances de classes représente une relation de dépendance entre classes.

2.2 Vues d'architecture

2.2.1 Méta-modélisation et vues associées au cadre de l'urbanisme

2.2.1.1 Définition des vues d'architecture

Les vues d'architecture d'un système dans le cadre de l'urbanisme sont la vue fonctionnelle, la vue technique et la vue applicative [Longépé, 2001]. Par rapport au modèle des vues dit "4+1" [Kruchten, 1995] :

- la vue fonctionnelle correspond à la vue des scénarios et à la vue logique,
- la vue technique correspond à la vue physique,
- la vue applicative correspond à la vue développement et à la vue processus.

La méta-modélisation permet de modéliser les concepts décrivant les vues associées au cadre de l'urbanisme ainsi que leurs liens. L'intérêt de ce méta-modèle est que chaque modèle d'architecture du SI ou d'un système du SI est une instantiation du méta-modèle [Kleppe, 2003]. L'objectif de réutilisation d'éléments du SI du niveau modèle est ainsi facilité par le partage, au niveau méta-modèle, des concepts permettant de décrire ces éléments.

Les vues associées au cadre de l'urbanisme sont les vues d'architecture d'un système auxquelles la vue métier supportée par le SI est ajoutée. L'ensemble des vues recouvre tous les concepts utiles à la description d'un SI et de ses systèmes :

- une description du métier de l'entreprise dans la vue métier,
- une description des fonctions du SI dans la vue fonctionnelle,
- une description des applications du SI dans la vue applicative,
- une description des technologies de déploiement des applications du SI dans la vue technique.

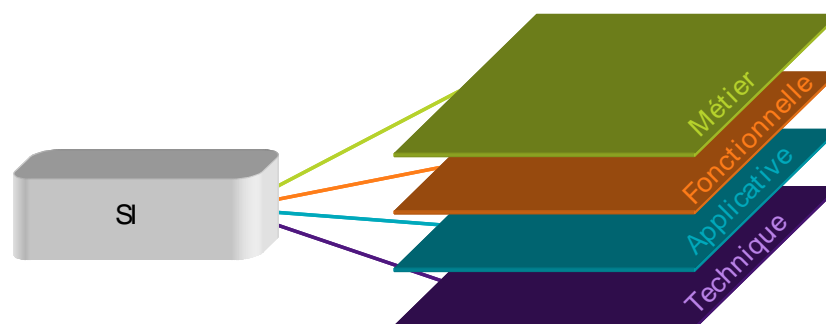


Figure 9 : Vues du cadre d'urbanisme d'un SI

Par rapport au développement d'un système du SI, la vue métier indique le "Pourquoi ?" de ce développement, la vue fonctionnelle précise le "Quoi ?", la vue technique décrit le "Avec quoi ?" et la vue applicative spécifie le "Comment ?".

Soit un processus de la vue métier lié à l'usage d'un logement, il serait supporté :

- par une vue d'architecture fonctionnelle recouvrant les fonctions (*se laver et se coucher*), et,
- par une vue d'architecture technique avec l'infrastructure choisie (*salle de bains, chambre à coucher*).

La vue d'architecture applicative du système de type SOA représentant le logement recouvrirait :

- la réalisation sous forme de service de ces fonctions intégrant un déploiement sur cette infrastructure (*se laver dans la salle de bains et se coucher dans la chambre*).

2.2.1.2 Vue métier

La vue métier décrit les processus du cœur de métier de l'entreprise et leur structuration par rapport à l'organisation de l'entreprise. Cette vue permet de référencer chaque élément du SI par rapport à un processus métier. Les concepts de la vue métier décrits dans le Tableau 1 sont repris des travaux de définition de l'analyse d'un processus métier dirigés par Francis Alizon à France Télécom R&D. Ces concepts sont dérivés des notions UML d'évènement, de processus, de tâche, d'acteur, d'activité ou de classe [Booch, 1999].

Concept	Définition
Événement métier	Notification par un processus métier d'une information à un autre processus métier, ou événement émis ou reçu par un acteur métier ou un collaborateur métier. Sa signification est directement perceptible par les différents acteurs individuels de l'entreprise. Il peut être porteur d'informations ou de matière.
Processus métier	Séquence d'actes réalisée par l'entreprise qui produit un résultat dont la valeur est perceptible et mesurable pour un acteur individuel de l'entreprise. Il doit être défini indépendamment de toute organisation et de tout système existant dans l'entreprise.
Tâche métier	Partie de traitement réalisée dans un processus métier, qui d'une part est définie par les informations (entités métier) qu'elle utilise en entrée et qu'elle produit en sortie, par les événements métier reçus et produits, et qui d'autre part respecte certaines règles : l'activité métier ne met en œuvre qu'une action principale, elle n'est pas interruptible.
Entité métier	Représentation d'un concept manipulé par l'entreprise dans son activité. Objet produit ou utilisé par un ou plusieurs processus métier.

Collaborateur métier	Représentation d'un rôle ou d'un ensemble de rôles joué dans la réalisation des processus métier par un individu ou une unité organisationnelle, interne au domaine modélisé (par exemple, toute l'entreprise). Un collaborateur métier contribue à la réalisation d'un ou plusieurs processus métier (il a un rôle pour chaque processus métier auquel il contribue), et plusieurs collaborateurs métier peuvent contribuer à la réalisation d'un processus métier. Un collaborateur métier émet ou reçoit des événements métier.
Procédure métier	Séquence d'actes réalisée par l'entreprise qui produit un résultat dont la valeur est perceptible et mesurable pour un acteur individuel de l'entreprise. Elle doit être définie dépendamment de l'organisation de l'entreprise ou de tout système existant dans l'entreprise.
Activité métier	Partie de traitement réalisée dans une procédure métier, qui d'une part est définie par les informations (données métier) qu'elle utilise en entrée et qu'elle produit en sortie, par les événements métier reçus et produits, et qui d'autre part respecte certaines règles : l'activité métier ne met en œuvre qu'une action principale, elle n'est pas interruptible, et elle est réalisée par un collaborateur métier unique à un moment donné.
Donnée métier	Représentation d'un concept manipulé par l'entreprise dans son activité. Objet produit ou utilisé par une ou plusieurs procédures métier.

Tableau 1 : Concepts de la vue métier.

La procédure métier est ici le résultat de la transformation d'un processus métier contraint par l'organisation de l'entreprise [Simonin, 2004]. Chaque tâche du processus est en effet transformée en autant d'activités qu'il y a de collaborateurs métier réalisant une partie de la tâche.

2.2.1.3 Vue fonctionnelle

Un extrait du méta-modèle de la vue fonctionnelle avec les concepts du PLU (cf. §2.1.4) est proposé dans la **Figure 10**. Le concept de donnée logique manipulée par un îlot fonctionnel (cf. Tableau 2) est intégré dans cet extrait du méta-modèle. La relation entre une donnée logique et l'îlot qui la produit ou l'utilise est en effet essentielle dans la vue fonctionnelle d'un système.

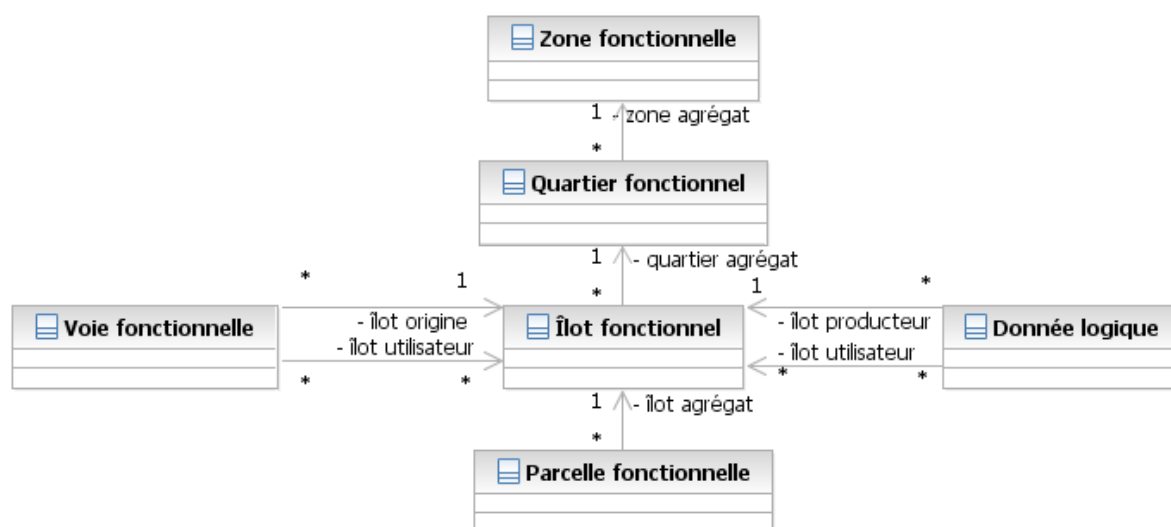


Figure 10 : Méta-modèle de la vue fonctionnelle enrichie du PLU

Les concepts de la vue fonctionnelle sont définis dans le Tableau 2 à partir :

- des éléments constitutants du PLU définis dans le §2.1.4,
- des résultats de travaux internes à France Télécom R&D dirigés par Francis Alizon sur l'analyse fonctionnelle d'un système [Nicolas, 2002]. Les termes UML de cas d'utilisation, de scénario, de classes et de relation entre classes [Booch, 1999] sont appliqués à la vue fonctionnelle.

Concept	Définition
Cas d'utilisation	Représentation d'une séquence d'actions d'un système faisant intervenir un ou plusieurs acteurs et produisant un résultat mesurable. Correspond à une grande fonctionnalité d'un système. Un système est caractérisé par l'ensemble de ses cas d'utilisation.
Scénario	Séquence d'interactions entre objets qui illustre un déroulement particulier d'un cas d'utilisation.
Entité	Objet utilisé ou produit par les cas d'utilisation ; c'est soit une notion métier, décrite au niveau de détail nécessaire et suffisant pour les besoins du système, soit une notion propre au système modélisé, dont l'existence est généralement due à la prise en compte de l'organisation.
Exigence fonctionnelle	Traduction d'un besoin fonctionnel d'un système.
Interaction entre entités	Ensemble d'échanges entre plusieurs entités pour offrir un service.
Parcelle fonctionnelle	Fonction élémentaire qui est le niveau de description fonctionnelle le plus bas d'un îlot fonctionnel.
Îlot fonctionnel	Regroupement de parcelles fonctionnelles qui est le plus bas niveau de découpe du PLU possédant une vue externe.
Quartier fonctionnel	Regroupement d'îlots fonctionnels homogènes quant à la nature de l'information fonctionnelle traitée au sens large.
Zone fonctionnelle	Regroupement de quartiers fonctionnels qui est le plus haut niveau de découpe du PLU.
Interaction entre îlots fonctionnels	Ensemble d'échanges entre plusieurs îlots fonctionnels réalisant l'usage d'un système.
Donnée logique	Objet utilisé ou produit par les îlots fonctionnels ; c'est un objet relatif aux entités définies par l'utilisation d'un système.
Voie fonctionnelle	Vue externe d'une relation entre îlots fonctionnels offrant un accès à des données logiques ou à des traitements sur ces données.

Tableau 2 : Concepts de la vue fonctionnelle.

Certains concepts décrivant une vue fonctionnelle sont illustrés dans le SI des architectures de réseau de télécommunication [Menai, 2005]. Les éléments fonctionnels du domaine réseau sont en effet des îlots fonctionnels alors que les points de référence reliant ces éléments fonctionnels correspondent aux voies fonctionnelles.

La définition des données logiques du réseau [Menai, 2006] est fondée sur un signal échangé entre deux éléments fonctionnels, soit une interaction entre îlots fonctionnels. Les deux éléments fonctionnels sont :

- l'élément fonctionnel émetteur du signal, soit l'îlot fonctionnel à l'origine de l'interaction,
- l'élément fonctionnel récepteur du signal, soit l'îlot fonctionnel cible de l'interaction.

Cette définition de la donnée logique, spécifique au réseau, est intéressante car la donnée est considérée comme le résultat d'une requête effectuée auprès d'un îlot fonctionnel lors d'une interaction réalisant un usage du système. Dans le cas général du développement d'un système, la donnée logique est modélisée de façon statique telle que dans le Modèle Logique de Données conçu avec la méthode Merise [Tardieu, 1985].

2.2.1.4 Vue technique

La vue technique décrit les moyens d'infrastructure utiles au déploiement des systèmes du SI. Les concepts de la vue technique décrits dans le Tableau 3 sont conformes aux termes d'exigence, de nœud et de connexion entre nœuds définis dans UML [Booch, 1999].

Concept	Définition
Exigence non fonctionnelle	Traduction d'un besoin technique ou opérationnel d'un système.
Nœud d'exécution	Machine physique ou environnement d'exécution permettant de supporter l'exécution des systèmes.
Protocole	Caractérisation d'un lien de communication qui permet le transfert d'information entre nœuds d'exécution.

Tableau 3 : Concepts de la vue technique.

2.2.1.5 Vue applicative

La vue applicative représente les composants applicatifs du SI et leurs relations de dépendance permettant de réaliser un système. Les concepts de la vue applicative du Tableau 4 sont déduits des notions UML de composant, d'interface et de relation entre composants [Booch, 1999].

Concept	Définition
Composant applicatif	Réalisation de fonctions présentant une finalité fonctionnelle ou une finalité technique décrite dans des cas d'utilisation. Il produit des données physiques fournies à d'autres composants organiques via ses interfaces. Un composant applicatif peut être l'agrégation de composants applicatifs.
Interaction entre composants applicatifs	Ensemble d'échanges entre plusieurs composants applicatifs pour offrir un service.
Interface applicative	Vue externe d'un composant applicatif offrant un accès à des données physiques ou à des traitements sur ces données.
Donnée physique	Objet utilisé ou produit par les composants organiques qui est la traduction d'une donnée logique sur un nœud d'exécution.

Tableau 4 : Concepts de la vue applicative.

Un concept de la vue applicative est aussi illustré dans le SI des architectures de réseau de télécommunication [Menai, 2005]. Il s'agit du concept d'organe qui est associé à celui de composant applicatif.

2.2.1.6 Règle d'urbanisme

A chaque concept appartenant à une des quatre vues définies précédemment, une règle contraignant son instanciation peut être associée. Cette règle, dite règle d'urbanisme, doit représenter de manière simple une contrainte à respecter lors de toutes évolutions du SI [Longépé, 2001] [Urba-SI, 2003] [Bonne, 2004]. Pour toutes évolutions de l'architecture d'un système, les architectes doivent vérifier que leur architecture respecte ces règles communes à l'entreprise. Par exemple, le fait qu'une donnée référentielle du SI doit être mise à jour par un seul système du SI est une règle d'urbanisme.

Définition : Une règle d'urbanisme est une règle paramétrée par des concepts extraits des méta-modèles décrivant chaque vue.

Les deux règles d'urbanisme suivantes qui ciblent les données produites illustrent cette définition. Ces règles définies à France Télécom R&D pour les services télécom sont déduites de règles d'urbanisme communes à de nombreuses entreprises qui concernent les données référentielles du SI [Urba-SI, 2003]. Ces règles doivent assurer à l'entreprise qui les applique une fiabilisation des données de son SI.

Règle d'urbanisme – Donnée produite et îlot fonctionnel n°1

Une donnée logique de la vue fonctionnelle n'est produite que par un seul îlot fonctionnel.

Règle d'urbanisme – Donnée produite et composant applicatif n°2

Une donnée physique de la vue applicative n'est produite que par un seul composant applicatif.

La première règle d'urbanisme est paramétrée par le concept de donnée logique et par le concept d'îlot fonctionnel décrit dans le méta-modèle de la vue fonctionnelle. La seconde règle d'urbanisme est paramétrée par le concept de donnée physique et par le concept de composant applicatif de la vue applicative.

Une règle d'urbanisme peut aussi représenter une règle de développement ou une règle liée au langage utilisé, que ce soit un langage de modélisation ou un langage de programmation. Par exemple, une règle définissant un scénario de la vue fonctionnelle comme une illustration d'un cas d'utilisation est totalement conforme à la sémantique UML.

Une règle d'urbanisme peut être représentée par une cardinalité dans une association d'un diagramme de classes UML représentant les concepts du méta-modèle des vues. C'est le cas de la Règle d'urbanisme – Donnée produite et îlot fonctionnel dans la **Figure 11** où la cardinalité de l'association entre la donnée logique et l'îlot fonctionnel est de 1 pour le rôle *îlot producteur*.

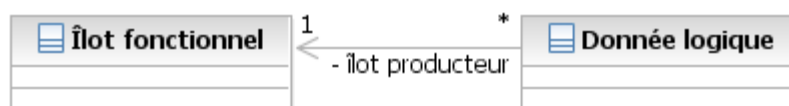


Figure 11 : Règle d'urbanisme – Donnée produite et îlot fonctionnel

La même règle peut être implémentée dans des langages liés à l'ingénierie des modèles tels que ceux proposés dans KerMeta [Muller, 2005], ATL [Jouault, 2006] ou SmartQVT [SmartQVT, web]. En OCL (Object Constraint Language) [Warmer, 1999], la Règle d'urbanisme – Donnée produite et îlot fonctionnel codée pour la classe UML *Donnée logique* est la suivante :

```

Context Donnée logique
inv self.îlot producteur->size()=1;
  
```

La liste des règles d'urbanisme proposées dans ce document est dans la table du chapitre 11.

2.2.2 Alignement entre la vue métier d'une entreprise et les vues du SI

La définition classique de l'alignement fondée sur l'EA est celle de l'alignement du point de vue métier d'une entreprise avec son SI. Cette définition reste néanmoins peu précise [Etien, 2006] alors que l'étude de l'alignement entre modèles est devenue nécessaire aux architectes décrivant le résultat de leur conception à l'aide de modèles [Chen, 2006]. L'alignement à partir de la stratégie de l'entreprise est décliné sur le métier et sur le SI en intégrant les processus métier et les rôles dans l'organisation de l'entreprise. L'alignement centré sur la stratégie de l'entreprise permet dans le cas de l'EA de comparer la prise en compte de cette stratégie au niveau métier et dans le SI [Henderson, 1993]. L'alignement entre le point de vue métier d'une entreprise et son SI est aussi le principe des SIS (Strategic Information System) où le cœur de métier intègre la structure de l'entreprise et sa stratégie [Wisemann, 1988]. Ce type d'alignement peut prendre en compte de façon complémentaire les concepts utiles au processus de définition d'un objectif et au moyen de l'atteindre tels que définis par l'EA [Rolland, 1999]. Le choix des concepts qui paramètrent cet alignement est spécifique à chaque entreprise [Luftman, 1999].

L'intégration de l'EA dans un processus de développement permet le partage d'un cadre fonctionnel, d'un cadre technique ou d'un cadre applicatif au niveau d'un SI. Cette intégration bénéficie à la réutilisation d'éléments de ce SI, caractéristique de sa durabilité. L'approche SEAM [Wegmann, 2003] qui intègre ces cadres dans le processus de développement d'un système a pour objectif d'améliorer l'alignement entre le métier de l'entreprise et le SI. L'amélioration est due à une intégration multidisciplinaire (marketing, cœur de métier de l'entreprise, SI) dans l'équipe de développement d'un système du SI et au partage des concepts ciblés par l'alignement [Wegmann, 2007]. Il est intéressant de noter dans SEAM que le modèle d'entreprise est une hiérarchie de systèmes. Ce principe hiérarchique est en effet proche de celui de la structure du PLU fonctionnel du SI (cf. §2.1.4). Cet alignement est difficilement mesurable car il reste indépendant d'une modélisation des concepts décrivant l'architecture fonctionnelle ou l'architecture technique d'un système.

L'alignement défini dans le Business Motivation Model [BRG, 2005] est spécifique au point de vue métier. Ce standard de l'Object Management Group est un méta-modèle décrivant les concepts du point de vue métier de l'EA (stratégie, objectif, bilan, etc.). L'objectif est de contrôler l'alignement entre le concept d'objectif et les autres concepts du méta-modèle du point de vue métier. Cet alignement est évalué à partir de la part de marché, des temps d'exécution d'un processus métier, de la satisfaction du client, etc. Ce type d'évaluation de l'alignement entre le point de vue métier de l'entreprise et le SI peut être réalisé avec des heuristiques [Pereira, 2005]. Chaque heuristique permet d'alerter l'architecte d'entreprise en cas de désalignement lié aux notions de risque, de coût ou de bénéfice associés au SI. De même que pour l'approche SEAM, une limitation est l'absence de mesures d'alignement entre l'architecture de la vue métier de l'entreprise et les architectures décrivant un SI et ses systèmes.

Une mesure de cet alignement peut être assimilée à une mesure de couplage [Etien, 2006]. L'alignement de concepts est en effet assimilable à un couplage de ces concepts. De nombreuses mesures de couplage existent dans le mode orienté objet [Chidamber, 1994]. Parmi ces mesures de couplage, certaines sont pertinentes dans le cadre d'une approche dynamique du développement de système car elles ciblent les interactions entre classes UML [Briand, 1999]. Les mesures appliquées aux modèles de données [MacDonell, 1997] complètent ces mesures d'alignement. La définition de mesures d'alignement corrélées au processus de développement d'un système reste cependant complexe [Kitchenham, 1995] [Briand, 1996].

Afin de la simplifier, le comportement de la mesure peut être axiomatisé [Fenton, 1986]. Des axiomes sont définis à partir des propriétés que doit vérifier la mesure. Ces propriétés se fondent sur l'intuition du comportement de la mesure lors de différentes hypothèses sur le contexte dans lequel la mesure est effectuée. Une fois définie, la mesure doit alors vérifier les axiomes spécifiant ce comportement.

2.3 Développement d'un système

2.3.1 Processus de développement UP

Dans une entreprise, le processus de développement d'un système satisfait généralement des normes de qualité telles que celles de l'ISO [ISO – 12207] [ISO – 15504]. C'est le cas du processus de développement UP [Jacobson, 1999]. Les vues d'architecture (cf. §2.2.1) propres au développement d'un système, soient les vues fonctionnelle, technique et applicative, sont intégrées aux macro-activités du processus de développement UP dans la **Figure 12**.

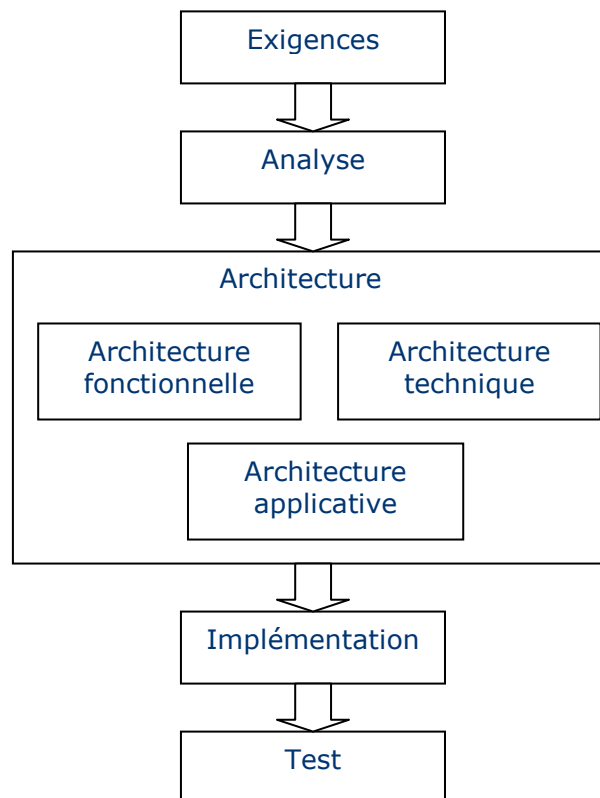


Figure 12 : UP et vues d'architecture

Le processus de développement UP est un processus itératif et incrémental. Une itération cible une macro-activité alors qu'un incrément cible un ensemble de macro-activités qui se succèdent dans le processus. Cette propriété du processus UP permet de définir des jalons décisionnels du type "STOP / ENCORE" lors du développement incrémental d'un système tel que décrit dans le modèle en spirale classique [Boehm, 1988].

La notion d'incrément est particulièrement importante dans un processus de développement. Il permet d'impliquer des acteurs contributeurs de macro-activités différentes dans un même incrément. Ceci implique qu'ils deviennent des parties prenantes de la production liée à l'incrément. Un exemple pertinent est celui des macro-activités de collecte des exigences et d'analyse. La collecte des exigences est réalisée pour un service télécom par des acteurs du marketing alors que l'analyse est faite par des analystes d'une équipe de développement. L'inclusion dans un même incrément de ces activités favorise la cohérence de l'analyse par rapport aux exigences collectées.

L'intérêt d'une macro-activité d'architecture itérative est, par exemple, d'enchaîner :

- pour la vue fonctionnelle, la conception d'îlots fonctionnels et la conception de données logiques manipulées par ces îlots, ou,
- pour la vue applicative, la conception de composants applicatifs et la conception de données physiques manipulées par ces composants.

L'apport de l'aspect itératif est d'assurer une meilleure cohérence globale du modèle d'architecture fonctionnelle ou du modèle d'architecture applicative.

Le Tableau 5 définit le modèle produit lors de chaque macro-activité du processus UP [Jacobson, 1999] intégrant les vues d'architecture (cf. §2.2.1).

Macro-activité	Modèle
Exigences	Modèle de cas d'utilisation
Analyse	Modèle d'analyse
Architecture fonctionnelle	Modèle d'architecture fonctionnelle
Architecture technique	Modèle d'architecture technique
Architecture applicative	Modèle d'architecture applicative
	Modèle de déploiement
Implémentation	Modèle d'implémentation
Test	Modèle de test

Tableau 5 : Macro-activités UP et modélisation.

Il est intéressant de noter l'ajout de l'EA, entre autres disciplines, à RUP dans EUP (Enterprise Unified Process) [Ambler, 2005]. L'objectif de cet ajout est de proposer des modèles d'architecture de référence aux architectes de système. Cette architecture de référence est définie comme un patron avec des artéfacts permettant leur usage.

Certaines macro-activités du processus de développement UP possèdent un aspect statique et un aspect dynamique. C'est le cas classique de l'analyse définie par un aspect statique et par un aspect dynamique dans les techniques de modélisation objet [Rumbaugh, 1991]. Dans [Roques, 2004], l'aspect statique de la macro-activité d'analyse cible les classes UML alors que l'aspect dynamique recouvre leur instantiation lors du déroulement d'un scénario décrivant l'usage d'un système. Globalement, les cas d'utilisation d'un système sont à l'origine des aspects statiques des macro-activités et les scénarios illustrant ces cas d'utilisation sont à l'origine de l'aspect dynamique des macro-activités.

2.3.2 Aspect dynamique d'un système

La vue privilégiée pour guider les travaux d'architecture d'un système de type architecture orientée service (SOA : Service Oriented Architecture) est la vue métier telle que définie par le Business Process Management (BPM) [Burlton, 2001]. Pour une entreprise qui fournit des services de télécommunication, cette vue contient par exemple les procédures d'utilisation d'un service télécom tel qu'un service de téléphonie, un service de messagerie ou un service de vidéo à la demande [Bertin, 2007]. L'entreprise construit un SI et ses systèmes afin de contribuer à la réalisation de chaque processus de son cœur de métier (cf. §1.1). Afin de réaliser un processus, la démarche est de concevoir un ensemble de séquences strictement ordonnées d'appels à des services fournis par les systèmes composant le SI. Par exemple, la réalisation d'un processus de commande de produit est réalisée par un appel à un service de création d'une commande, puis un appel de création d'un client et enfin un appel de recherche d'un article dans le catalogue de produits. Une lacune dans cette vue qui décrit les processus du cœur de métier de l'entreprise est l'absence de l'aspect dynamique du SI [Kaisler, 2005]. La vue métier ne peut en effet représenter les flux entre les systèmes du SI.

Afin d'illustrer la difficulté de définir de manière optimale les services propres à la réutilisation dans une architecture SOA, l'architecture d'un système est représentée ci-après par l'analogie avec l'usage d'un logement comme processus métier et avec un logement à réaliser comme système du SI. La réalisation de ce processus métier équivaut à accéder à une séquence de services où chaque service est déployé dans une des pièces du logement. L'accès à un service se fait à partir d'un couloir qui joue le rôle de bus de service en offrant une possibilité d'accès à tous les services déployés dans chaque pièce du logement.

Par exemple, la séquence temporellement ordonnée de services fournis par le logement décrit **Figure 13** réalise le processus métier *se coucher dans un logement* :

1. *se laver dans la salle de bains,*
2. *se coucher dans la chambre à coucher.*

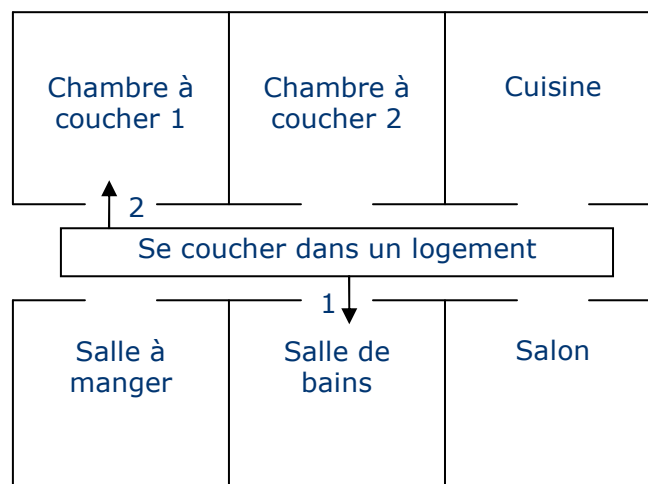


Figure 13 : Processus *se coucher dans un logement* et séquence ordonnée de services

Le principal apport de l'aspect dynamique est de reposer sur l'étude des dépendances entre fonctions. Dans l'illustration du logement, la fonction *se coucher* peut dépendre directement de la fonction *se laver*. Pour l'architecte, cette dépendance signifie que l'affirmation "*se coucher* nécessite de *se laver*" est plus cohérente avec le processus d'usage du logement que la séquence ordonnée représentée dans la **Figure 13**. L'architecte peut en déduire un accès indirect au service de la vue applicative *se laver dans la salle de bains* via le service *se coucher dans la chambre 1*, et non plus un accès direct.

Les conséquences sur la vue applicative représentées **Figure 14** sont une utilisation optimisée du couloir du fait de l'accès indirect à la salle de bains. Le passage dans le couloir, assimilé dans l'allégorie à un bus de services [Robinson, 2004], est ainsi diminué. Ce bus joue un rôle de médiateur entre l'utilisateur de la maison et la maison. De plus, la vue technique est optimisée en positionnant la salle de bains entre les deux chambres afin de déployer dans chaque chambre la réalisation de cette dépendance fonctionnelle.

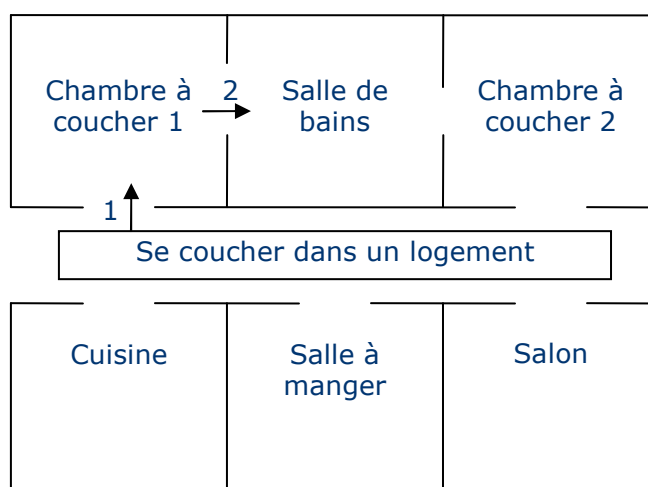


Figure 14 : Processus *se coucher dans un logement* et dépendance de services

Le processus orienté client est une solution permettant de répondre à l'évolutivité des marchés et des technologies. La méthode BPR (Business Process Reengineering) [Schael, 1997], issue de la réingénierie [Hammer, 1993], permet de modéliser les processus orientés client incontournables pour l'entreprise. La réponse technologique est le fonctionnement en réseaux coopératifs afin de répondre à la demande du client. Ce fonctionnement nécessite néanmoins pour Thomas Schael une amélioration au niveau fonctionnel.

L'aspect dynamique apparaît aussi dans le modèle architectural en couches inspiré du modèle OSI (Open Systems Interconnect) [Zimmermann, 1980]. Ce modèle en couches est appliqué en informatique aux architectures N-tier. La localisation sur une couche de chaque élément de l'architecture d'un système permet de mettre en œuvre des règles d'interaction entre éléments de couches différentes. La structure en couches stabilise ainsi une architecture.

La couche de présentation d'un système, ou IHM (Interface Homme-Machine), est la plus impactée par l'aspect dynamique d'un système. Cette couche de présentation est spécifiée dynamiquement par une cinématique des écrans offerts à l'utilisateur du système [Armour, 2001]. Les scénarios d'usage d'un système sont ainsi classiquement utilisés dans les méthodes de conception des IHM [Carroll, 1995]. Les interactions successives entre un utilisateur d'un système et le système doivent en effet représenter un usage spécifié du système. La réutilisation de composants peut aussi bénéficier de la décomposition a posteriori de scénarios d'usage d'un système [Rolland, 1998].

2.3.3 Développement d'un système et approche MDA

La modélisation est une solution à la complexité dans l'approche systémique présentée dans l'introduction (cf. §1.1), que ce soit la complexité d'un SI ou la complexité d'un système. Le choix de l'urbanisme comme cadre de l'EA permet de modéliser de façon hiérarchique les vues d'un SI grâce aux concepts du PLU. Ces mêmes vues appliquées à un système de ce SI doivent donc aussi être modélisées afin d'être simplifiées. Le choix de l'ingénierie des modèles permet de concevoir des solutions de modélisation de l'architecture d'un système en prenant en compte la contrainte de modèles des vues d'un SI cible.

La conception dirigée par les modèles (Model Driven Design) permet en effet de développer des systèmes à partir de modèles décrivant le comportement des systèmes représentés par des scénarios, donc par l'aspect dynamique de ces systèmes [Jézéquel, 2008]. Dans un processus de développement, l'approche d'ingénierie des modèles cible généralement l'architecture applicative détaillée, la génération de code utile à la macro-activité d'implémentation, et les tests. Pour l'architecture applicative détaillée, la projection de chaque composant applicatif sur une couche applicative (couche présentation, couche d'accès aux données, etc.) est détaillée à l'aide d'un environnement architectural propre à chaque couche. Les approches MDE [Frankel, 2003] ou MDA [MDA, web] favorisent l'intégration d'environnements architecturaux applicatifs durant un processus de développement de système [Guelfi, 2004]. Concernant les tests, l'approche MDA permet de tester les systèmes générés à partir de transformations de modèles en validant ces transformations [Fleurey, 2004].

L'approche MDA est une norme proposée par l'OMG (Object Management Group) qui cible la transformation de modèles. Dans le cadre proposé par l'OMG, la plate-forme est liée à un environnement d'exécution (Java, Corba, etc.). Les modèles impactés sont le PIM (Platform Independent Model), le PSM (Platform Specific Model) et le modèle de la plate-forme [Miller, 2003]. L'utilisation méthodologique de ce modèle de plate-forme sous le sigle de PDM (Platform Description Model) est définie par le fait que le PSM est une configuration du PIM pour un PDM donné [Gervais, 2002].

L'objectif de l'approche MDA est de s'affranchir de l'hétérogénéité des plates-formes d'exécution. L'idée représentée dans la **Figure 15** est de transformer un modèle indépendant de la plate-forme (PIM) en un modèle spécifique à cette plate-forme (PSM) en contraignant cette transformation par un modèle définissant la plate-forme (PDM). Cette plate-forme peut aussi être prise en compte directement dans le PSM, c'est-à-dire sans transformation d'un modèle indépendant des contraintes de la plate-forme. Un autre modèle est défini dans l'approche MDA pour représenter les exigences fonctionnelles du système à développer. Ce modèle est le CIM (Computation Independent Model). Ce modèle peut être transformé en un PIM. Cette transformation non contrainte représente le passage des exigences vers l'analyse et la conception définies dans UP. La transformation du PIM vers le PSM représente dans l'approche MDA le passage de la conception générale du système à la conception détaillée, dernière étape avant le codage d'une application [Blanc, 2005].

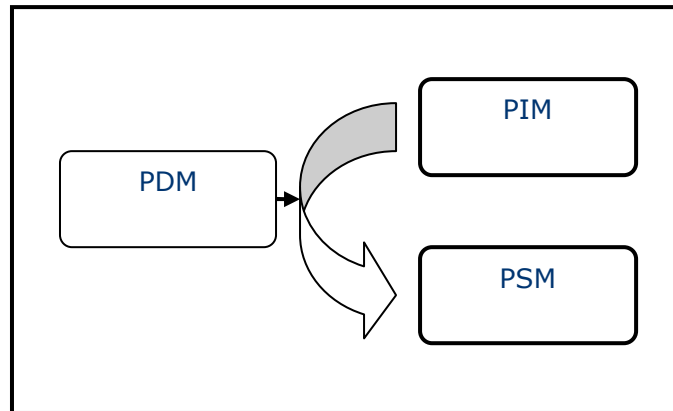


Figure 15 : PIM, PDM et PSM dans l'approche MDA

Les transformations de modèle ont pour paramètres les concepts de chaque vue du SI (cf. §2.2.1). Les méta-modèles utiles à chaque transformation, soient ceux du PIM, du PSM et du PDM, sont extraits des méta-modèles des vues. Ces méta-modèles peuvent être représentés par des diagrammes de classes UML. Les transformations de modèles liées au développement d'un système peuvent ainsi être représentées par un diagramme de classes représentant les concepts impactés. L'ingénierie des modèles permet d'intégrer au mieux la méta-modélisation des concepts décrivant un système du SI [Menaï, 2005]. De plus, elle permet de représenter une macro-activité du processus UP par une transformation de modèle fondée sur les concepts définissant les différentes vues.

2.4 Conclusion

Une première difficulté soulignée dans l'état de l'art est la qualité de la conception des éléments à réutiliser dans le SI. Une solution à cette difficulté rentre dans le cadre du SOEA (Service Oriented Enterprise Architecture) [Grigoriu, 2007]. Les composants conçus par les architectes d'entreprise ou les urbanistes du SI sont implicitement les composants appropriés à une architecture SOA [Grigoriu, 2006]. La démarche DA4EA présentée dans la thèse offre une solution de définition des services à partir de la vue fonctionnelle de l'EA.

De même, la prise en compte par l'architecte d'un système des modèles réalisés par les architectes d'entreprise est complexe. La première raison est l'absence d'une modélisation pertinente de l'EA d'un SI. Une seconde raison est une prise en compte non automatisée de l'EA lors du développement d'un système. Cette intégration des composants de l'EA peut bénéficier du cadre de l'urbanisme d'un SI et de l'ingénierie des modèles. La démarche EA4UP décrite dans la thèse recouvre l'intégration de l'EA lors du développement d'un système. Dans cette démarche, l'approche MDA est choisie pour transformer un modèle indépendant d'une plate-forme en un modèle dépendant de cette plate-forme. La définition de la plate-forme, ou PDM, contraignant la transformation de modèles est étendue dans la démarche EA4UP.

Une complexité méthodologique est de plus induite par la modélisation en parallèle des données et des traitements telle que préconisée dans Merise [Tardieu, 1985]. Pour résoudre cette complexité, la démarche EA4UP est fondée sur l'aspect dynamique du système. La conception d'une séquence de fonctions réalisant un scénario d'usage d'un système précède en effet dans la démarche EA4UP une modélisation des données manipulées par chaque fonction durant ce scénario. La prise en compte plus directe de la dynamique des scénarios d'utilisation des systèmes lors de leur conception a pour objectif d'améliorer, au niveau fonctionnel, la réutilisabilité d'éléments du SI. La décomposition des scénarios proposée est de plus réalisée a priori par une définition de la vue fonctionnelle cible du SI. Les scénarios d'usage d'un système sont de ce fait des éléments importants en entrée des démarches DA4EA et EA4UP.

Une autre difficulté rappelée dans l'état de l'art est de définir précisément un alignement et d'en déduire une mesure. Des concepts du méta-modèle des vues de l'urbanisme sont sélectionnés afin de définir des alignements propres aux démarches DA4EA et EA4UP. Cette solution est d'abord proposée dans le cadre d'une mesure de l'alignement entre la vue métier d'une entreprise et la vue fonctionnelle d'un SI dans la démarche DA4EA. Dans le troisième point de la contribution de la thèse, cette solution est mise en œuvre pour une mesure d'alignement permettant d'évaluer l'écart entre la vue fonctionnelle urbanisée du SI et l'architecture fonctionnelle conçue pour un système de ce SI.

Cette mesure d'alignement fonctionnel est étudiée ici dans le cadre théorique de la théorie de l'information [Shannon, 1948]. Les voies entre éléments fonctionnels conçues dans la vue fonctionnelle du SI sont considérées lors de l'élaboration de la mesure à un canal de communication. Pour chaque scénario d'usage d'un système, la vue fonctionnelle du SI a alors pour rôle de transmettre chaque interaction du scénario. La qualité de la transmission dépend du niveau (zone, quartier, îlot) de la voie entre éléments fonctionnels utilisée par l'interaction. La mesure d'alignement choisie est l'information mutuelle moyenne définie par Claude Shannon qui évalue la qualité de la transmission. L'intérêt de cette approche est de définir une mesure mathématique d'un alignement dépendant des concepts du méta-modèle des vues du SI ou d'un système.

3. La démarche DA4EA d'urbanisation de la vue fonctionnelle d'un SI

3.1 Principes

Si l'illustration du logement à réaliser est reprise (cf. §2.3.2) pour présenter la démarche DA4EA, l'objectif serait d'assister l'architecte fonctionnel devant concevoir un logement modèle. La démarche DA4EA aide l'architecte de ce modèle fonctionnel :

- en stabilisant son architecture grâce à un typage des fonctions d'usage d'un logement (les fonctions *se coucher* et *aménager les combles* ne sont pas du même type dans le processus d'usage d'un logement puisque les cycles de vie des deux fonctions sont différents),
- en offrant un outil de regroupement de fonctions proches dédiées à l'usage d'un logement (un ensemble de fonctions dédiées au couchage, un ensemble de fonctions dédiées aux repas, etc.).

3.1.1 La vue fonctionnelle déduite de la vue métier ou alignée avec la vue métier ?

L'urbanisation de la vue métier d'une entreprise et l'urbanisation de la vue fonctionnelle d'un SI définies dans le §2.2.1.2 respectent toutes les deux la stratégie de l'entreprise. Ce respect commun définit un lien entre la vue métier supportant un SI et la vue fonctionnelle de ce SI. Dans le cadre d'une urbanisation, la vue fonctionnelle est généralement déduite de la vue métier. Cette vue fonctionnelle peut aussi être conçue de manière spécifique. Dans ce dernier cas, le lien entre les deux vues est concrétisé par l'évaluation de leur alignement.

La déduction d'une vue fonctionnelle à partir d'une vue métier est considérée comme une transformation, réalisée par des experts, des axes stratégiques métier vers la vue fonctionnelle ou vers la vue technique. Cette transformation est représentée dans [Longépé, 2001] par un enchaînement de diagrammes d'Ishikawa représentant les axes stratégiques de la vue métier ou de la vue fonctionnelle. La notion de transformation dans le cadre de l'urbanisme nécessite aussi de concevoir un lien explicite entre des concepts du méta-modèle décrivant chacune des vues. Chaque lien est complété par des règles d'urbanisme. Un exemple de règle pourrait être de déduire un îlot de la vue fonctionnelle à partir d'une tâche de la vue métier et d'une entité métier produite par la tâche.

Par exemple, pour les SI de France Télécom, un lien de déduction devrait être conçu entre les processus décrits dans eTOM [eTOM, web] par le TMF (TeleManagementForum), dont un extrait ciblant le client est proposé **Figure 16**, et la vue fonctionnelle des SI de France Télécom représentée dans la **Figure 17** pour le domaine lié au client de l'opérateur.

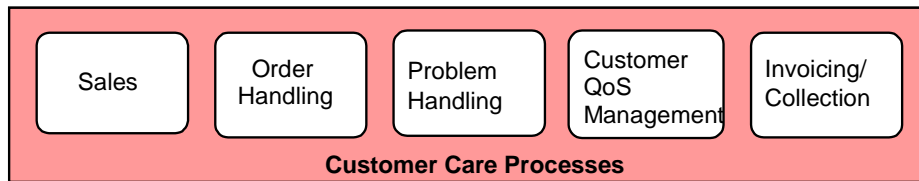


Figure 16 : Extrait de la vue métier eTOM

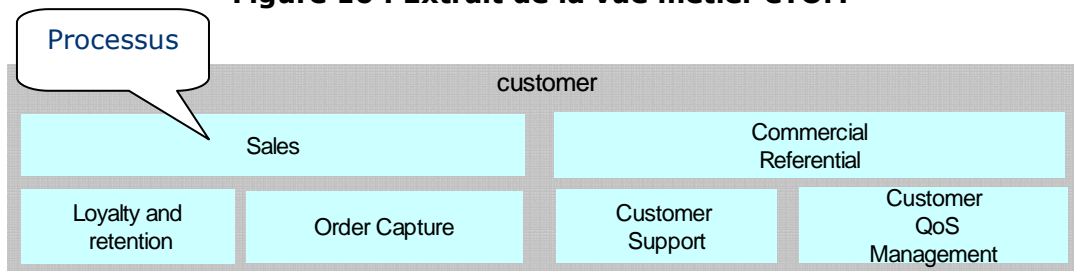


Figure 17 : Extrait de la vue fonctionnelle du SI de France Télécom

Le lien entre ces deux vues est immédiat entre le processus *Sales* d'eTOM et le bloc fonctionnel *Sales* de la vue fonctionnelle du SI de France Télécom. Par contre, le bloc fonctionnel *Commercial Referential* qui est déduit de l'ensemble des processus *Customer Care Processes* souligne l'importance de la gestion du client dans le SI de France Télécom. Cette importance est mise en relief par des applications spécifiques à la gestion du client. La vue fonctionnelle des SI de France Télécom est donc déduite directement de la vue métier tout en intégrant le périmètre fonctionnel réalisé par les applications de ses SI.

Dans le cas de la conception spécifique de la vue fonctionnelle d'un SI, la vue fonctionnelle est conçue par un urbaniste fonctionnel expert d'une zone fonctionnelle. L'urbaniste intègre directement la stratégie marketing ou les évolutions technologiques, lorsque le cœur de métier de l'entreprise le nécessite, lors de la conception de la vue fonctionnelle de sa zone.

La vue fonctionnelle du SI des services télécom de France Télécom est une illustration de cette conception spécifique. Les zones de cette vue fonctionnelle conçues par les urbanistes de France Télécom R&D sont représentées dans la **Figure 18**.

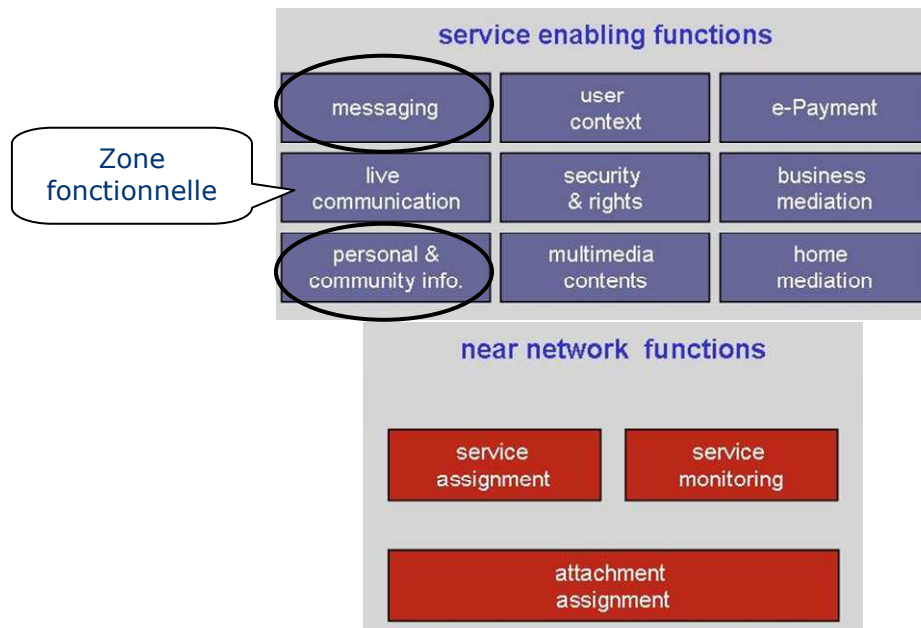


Figure 18 : Extrait de la vue fonctionnelle du SI des services télécom de France Télécom

Ces zones supportent les processus d'usage d'un service télécom. La **Figure 19** représente un de ces processus, soit le processus d'envoi de message électronique décrit dans un diagramme d'activités UML par Emmanuel Bertin lors de ses travaux à France Télécom R&D sur la conception de la vue métier liée à l'usage des services télécom.

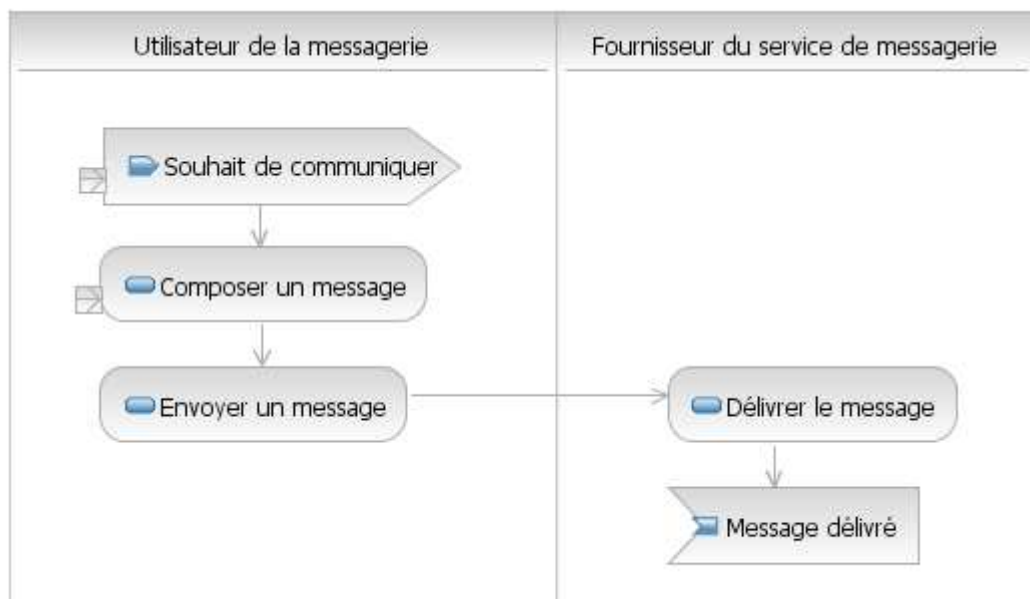


Figure 19 : Extrait de la vue métier du SI des services télécom de France Télécom

Les zones fonctionnelles du SI des services télécom entourées dans la **Figure 18** supportent le processus d'envoi de message électronique. Ces zones fonctionnelles sont la zone *Messaging* pour l'édition et l'envoi du message et la zone *Personal & community info* pour la gestion de l'adresse du destinataire appartenant au carnet d'adresse de l'expéditeur du message.

L'existence propre d'une vue fonctionnelle d'un SI nécessite néanmoins d'être alignée sur la vue métier de l'entreprise. Cet alignement, qui est un cas particulier de l'alignement du cœur de métier de l'entreprise avec le SI (cf. §2.2.2), représente en effet la cohérence de ces deux vues quant à la prise en compte de la stratégie de l'entreprise.

La conception d'une vue fonctionnelle d'un SI avec la démarche DA4EA proposée dans ce chapitre est une conception spécifique. Cette conception tient donc compte de l'alignement de la vue fonctionnelle avec la vue métier.

3.1.2 Description d'ensemble de la démarche DA4EA

La démarche DA4EA décrite **Figure 20** recouvre les activités suivantes :

- La conception par les urbanistes de chaque zone des îlots fonctionnels et de leurs parcelles, des voies de la vue fonctionnelle de la zone. Cette conception d'expert prend en compte la stratégie de l'entreprise ;
- Le typage des îlots fonctionnels réalisé par l'urbaniste, expert de la zone contenant les îlots. L'activité permet de créer, de valider ou d'invalider des voies entre îlots ;
- La mesure de l'alignement entre la vue fonctionnelle et la vue métier. Le bilan réalisé à partir de cette mesure permet de faire évoluer le périmètre fonctionnel des îlots ou de modifier les voies qui les relient ;
- La conception des quartiers fonctionnels fondée sur une approche dynamique ciblant les îlots fonctionnels conçus.

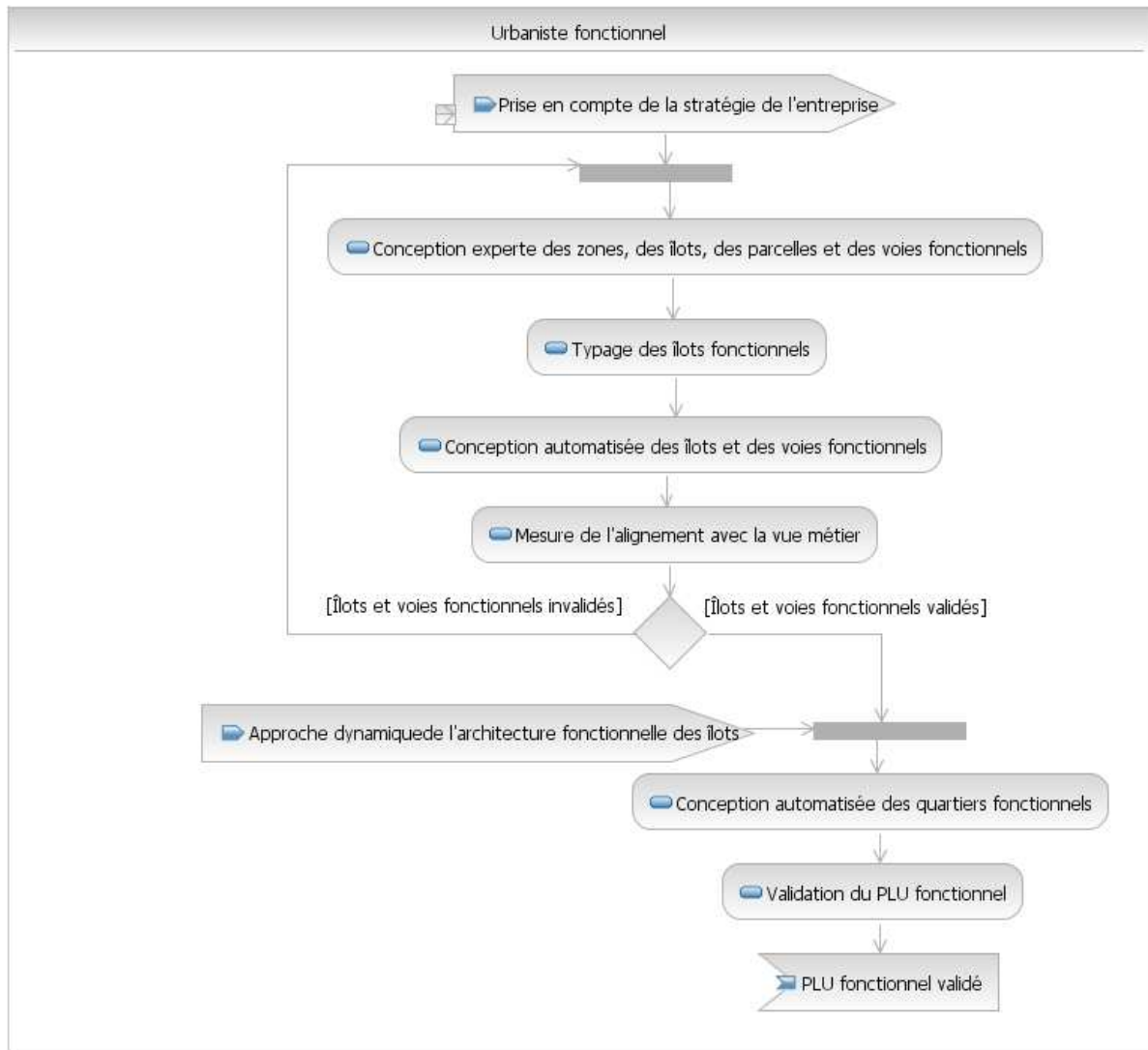


Figure 20 : Démarche DA4EA

L'ensemble des activités autour des îlots et des voies de la démarche DA4EA nécessitent une approche incrémentale (cf. §2.3.1). Grâce à la mesure de l'alignement de la vue fonctionnelle, l'urbaniste de la zone concernée peut :

- réajuster la définition des parcelles d'un îlot,
- ajouter ou supprimer une voie entre deux îlots.

Deux activités de la démarche DA4EA sont automatisables. La stabilisation des îlots et des voies à partir du typage des îlots et la conception des quartiers sont en effet le résultat de l'application de règles d'urbanisme. Afin d'implémenter ces règles, l'approche MDE [Frankel, 2003] et le langage QVT [QVT, web] sont choisis. L'intérêt de cette approche est de bénéficier du méta-modèle des vues d'urbanisme et des règles d'urbanisme qui lui sont associées pour créer la transformation de modèle.

3.2 Conception experte des zones et des îlots du PLU fonctionnel

La conception des zones et des îlots du PLU fonctionnel est une activité complexe de l'urbanisation. La complexité naît de l'intégration concrète de la stratégie de l'entreprise dans la vue fonctionnelle d'un SI qui est abstraite. Le contenu de cette activité entourée dans la **Figure 21** est décrit dans ce paragraphe sans être approfondi du point de vue méthodologique.

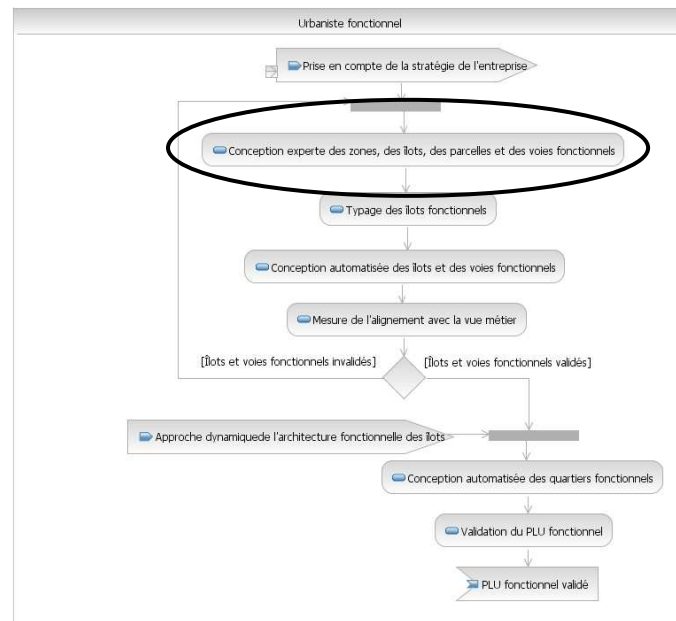


Figure 21 : Activité de conception experte des zones, des îlots et des voies fonctionnels dans la démarche DA4EA

La découpe en zones fonctionnelles supportant un processus métier est le résultat du travail d'un groupe d'experts de ce SI. La caractéristique importante de cette découpe est de ne prendre en considération que l'aspect fonctionnel du SI. La difficulté est de dissocier le fonctionnel, vu de l'utilisateur de ce SI, du fonctionnel dont l'utilisateur bénéficie de façon implicite. Par exemple, la fonction envoyer un message électronique est perçue de l'utilisateur alors que la fonction transmettre à haut débit bénéficie à l'envoi de son message, sans être explicite, a priori, à l'utilisateur.

L'impact de la stratégie de l'entreprise permet à ce groupe d'experts de souligner l'importance d'un axe fonctionnel stratégique en le recouvrant par une zone fonctionnelle. Dans l'illustration de la messagerie, un axe stratégique pourrait être de proposer à un utilisateur d'unifier ses messageries sur le réseau mobile, le réseau fixe et le réseau internet. La stratégie marketing serait alors d'offrir un service télécom permettant à un utilisateur de consulter ses messages, quelque soit le réseau auquel il est connecté. La création d'une zone messagerie intégrant l'unification soulignerait son importance pour l'entreprise.

La définition des îlots d'une zone fonctionnelle par l'urbaniste expert de la zone doit aussi être conforme à la stratégie de l'entreprise. Si la stratégie de l'opérateur de télécommunication est de proposer une sécurité renforcée lors de la réception d'un message électronique, l'urbaniste fonctionnel devrait souligner cet axe stratégique par la création d'un îlot fonctionnel de *sécurité de la messagerie*. Cet îlot comprendrait les parcelles suivantes : *Filtrer un message par rapport à l'objet du message*, *Filtrer le message par rapport à l'expéditeur du message*, *Créer les règles de filtrage de message*.

La conception de voies entre îlots est plus difficile pour l'urbaniste. Elle nécessite de prendre en compte la dépendance fonctionnelle d'un îlot vis-à-vis d'un autre îlot. Cette dépendance doit être cohérente avec son instantiation au sens UML dans une approche dynamique (cf. §2.3.2) de la conception de la vue fonctionnelle. La cohérence des voies fonctionnelles avec les interactions entre instances d'îlots fonctionnels durant un scénario d'utilisation du système est spécifiée dans la règle d'urbanisme suivante. La Règle d'urbanisme – Interactions et voies nécessite le typage de l'interaction entre deux îlots fonctionnels défini ci-après.

Définition : une interaction entre instances d'îlots fonctionnels est :

- de type <<requête>> si l'interaction est une demande, ou si l'interaction correspond à l'émission de paramètres d'entrée d'une requête,
- de type <<réponse>> si l'interaction est une réponse à une demande ou si l'interaction correspond à l'émission de paramètres de sortie d'une requête.

Règle d'urbanisme – Interactions et voies n°3

Chaque interaction entre instances d'îlots fonctionnels spécifiée à partir d'un scénario d'un cas d'utilisation d'un système doit être conforme avec une voie entre îlots fonctionnels :

- si l'interaction de type requête est adressée à une instance de l'îlot fonctionnel I_2 à partir d'une instance de l'îlot I_1 , elle nécessite une voie de l'îlot I_1 vers l'îlot I_2 .
- si l'interaction de type réponse est adressée à une instance de l'îlot fonctionnel I_1 à partir d'une instance de l'îlot I_2 , elle nécessite une voie de l'îlot I_1 vers l'îlot I_2 .

Dans les deux cas, l'îlot I_1 dépend de l'îlot I_2 .

Avec une modélisation formulée avec le langage UML, cette règle signifie que le diagramme de séquence représenté dans la **Figure 23** est conforme au diagramme de classe représenté dans la **Figure 22**.

Dans ces diagrammes :

- un îlot fonctionnel est représenté par une classe UML stéréotypée "îlot fonctionnel",
- une voie d'un îlot fonctionnel vers une autre est une dépendance UML stéréotypée "dépend de",
- une interaction de type requête est stéréotypée "type requête",
- une interaction de type réponse est stéréotypée "type réponse".



Figure 22 : Modèle d'îlots fonctionnels de référence

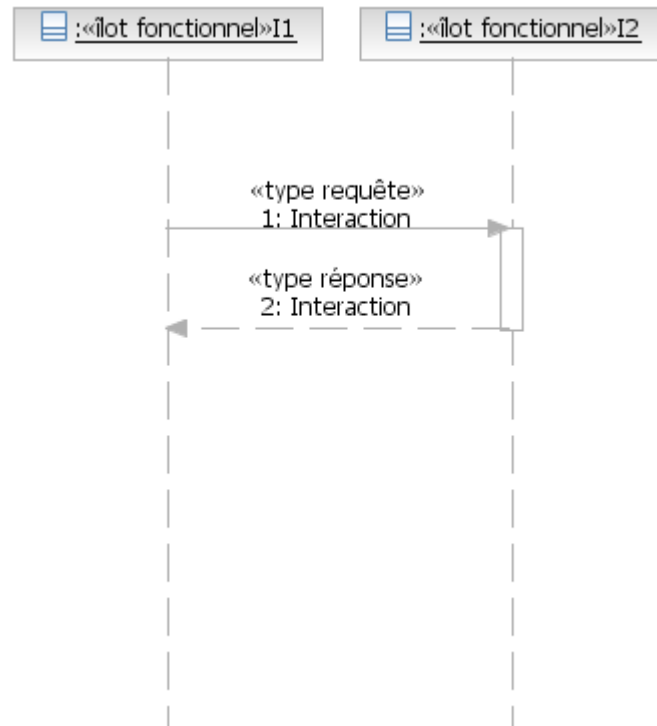


Figure 23 : Diagramme de séquence conforme à un modèle d'îlots fonctionnels

L'interaction de type requête est une opération au sens UML offerte par l'îlot cible de cette interaction. Cet îlot est aussi la source de l'interaction de type réponse déduite de la requête.

Supposons qu'un axe stratégique du marketing soit d'associer la réception d'un message à une offre de sécurité pour l'utilisateur qui reçoit le message. Du fait de cet axe stratégique, si l'urbaniste crée un îlot fonctionnel de gestion de la sécurité de la messagerie, l'îlot fonctionnel de *Sécurité de la messagerie* dépendrait de l'îlot de *Réception d'un message*. La sécurité de la messagerie est en effet responsable de la réception d'un message, et non l'inverse. La **Figure 24** représente dans un diagramme de classes UML cette dépendance. Dans ce diagramme de classes, une parcelle fonctionnelle est une méthode de la classe UML représentant l'îlot auquel elle appartient.



Figure 24 : Illustration de la dépendance d'îlots de la vue fonctionnelle d'un SI

Cette dépendance entre îlots est instanciée au sens UML lors d'un scénario du service télécom de messagerie électronique. Le diagramme de séquence UML de la **Figure 25** représente un scénario possible instanciant la dépendance entre les îlots *Sécurité de la messagerie* et *Réception d'un message*.

Dans le diagramme de séquence,

- une interaction de type requête ayant pour cible un îlot fonctionnel est libellée par l'intitulé de la parcelle utilisée dans cet îlot,
- une interaction de type réponse ayant pour source un îlot fonctionnel est libellée par l'intitulé de la parcelle utilisée dans cet îlot.

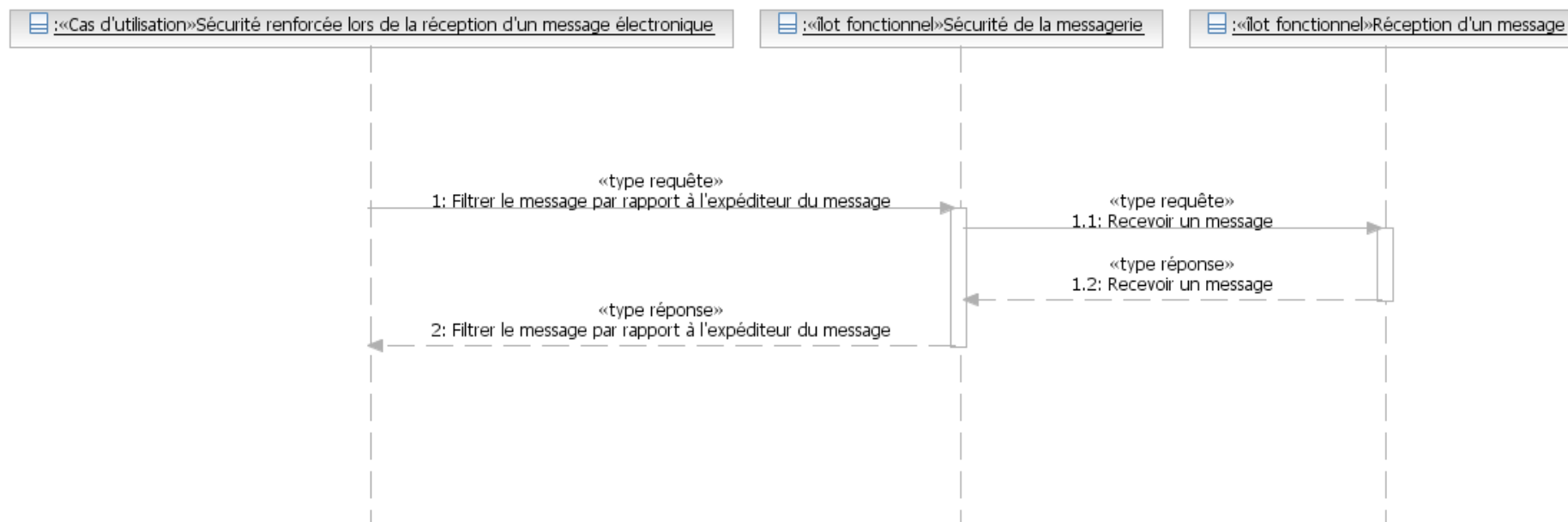


Figure 25 : Illustration d'un scénario instanciant des îlots dépendants de la vue fonctionnelle d'un SI

Le scénario de mise en œuvre de fonctions de sécurité lors de la réception d'un message est représenté par une interaction ayant pour source une instance de l'îlot fonctionnel *Sécurité de la messagerie* et pour cible une instance de l'îlot fonctionnel de *Réception d'un message*. Cette interaction succède à celle de filtrage d'un message puisque le filtrage est appliquée lors la réception du message.

3.3 Conception des îlots du PLU fonctionnel

Un premier objectif de la démarche DA4EA est d'assister l'urbaniste fonctionnel lors de l'activité de conception des voies entre îlots. La solution proposée est fondée sur le typage de chaque îlot fonctionnel par rapport à la durée de vie de son instance. Un modèle type des voies entre des îlots de type différent permet l'atteinte de l'objectif.

3.3.1 Définition du typage d'un îlot fonctionnel

Le principe du typage est issu du concept de couche architecturale (cf. §2.3.2). Le typage de chaque élément permet la localisation de l'élément sur une couche. Ce principe est appliqué à la conception des îlots fonctionnels et des voies qui les relient.

L'activité de typage d'un élément fonctionnel entourée dans la **Figure 26** peut caractériser :

- un îlot fonctionnel ou
- une donnée logique produite ou utilisée par un îlot fonctionnel [INPI – 04617] [INPI – 05928] [INPI – 06470].

Le typage d'un îlot fonctionnel ou d'une donnée de la vue fonctionnelle est commun respectivement à toutes ses parcelles ou à tous ses attributs. Le principe de ce typage est de considérer la dépendance de la durée de vie d'une instance au sens UML de l'élément fonctionnel par rapport à celle du processus supporté.



Figure 26 : Activité de typage des îlots fonctionnels dans la démarche DA4EA

Définition : le typage des îlots fonctionnels est le suivant :

- un îlot fonctionnel est de type flux si la durée de vie d'une instance de chacune de ses parcelles est dépendante du processus supporté ;
- Un îlot fonctionnel est de type stock si la durée de vie d'une instance de chacune de ses parcelles est indépendante du processus supporté.

Si T est l'ensemble des deux types d'îlots fonctionnels, si $P(T)$ est l'ensemble des parties de T , c'est-à-dire l'ensemble des sous-ensembles de T , et si I_f est l'ensemble des îlots fonctionnels du PLU, alors le typage t d'un îlot fonctionnel i peut être représenté de la manière suivante :

$$\begin{array}{l} t : I_f \rightarrow P(T) \\ i \mapsto t; t \subset P(\{\text{flux}, \text{stock}\}) \end{array}$$

Équation 3-1

La difficulté du typage d'un îlot est d'avoir un libellé non ambigu pour chacune de ses parcelles. Le choix d'une sémantique de référence telle que le modèle Create – Read – Update – Delete (CRUD) et des règles associées [Kilov, 1994] permet de clarifier le sens du verbe du libellé la parcelle. En ce qui concerne le complément du verbe, sa définition doit lever toutes ambiguïtés sur le libellé de la parcelle. Par exemple, la parcelle *Créer un message* correspond-elle à la création de la structure du message ou à sa date d'envoi ? Ce type d'interrogation sur l'ambiguïté des libellés des parcelles d'un îlot est fréquent durant toute la phase de typage d'un îlot fonctionnel.

L'objectif du typage des îlots est de stabiliser l'architecture fonctionnelle du PLU par un modèle en couches (cf. §2.3.2). Le typage des îlots fonctionnels a pour but d'associer chaque îlot à une couche flux ou à une couche stock de l'architecture fonctionnelle d'un système. Afin de stabiliser les voies entre îlots, la Règle d'urbanisme – Flux et stock contraint l'orientation des voies. La pertinence d'un typage d'un élément fonctionnel est en effet liée à l'existence d'un pattern de dépendance entre des éléments fonctionnels de type différent.

Règle d'urbanisme – Flux et stock n°4

Si un îlot fonctionnel de type flux et un îlot fonctionnel de type stock sont dépendants, alors l'îlot fonctionnel de type flux dépend de l'îlot fonctionnel de type stock.

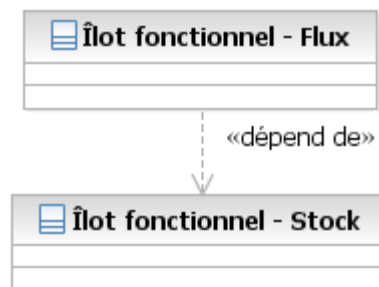


Figure 27 : Règle de dépendance d'îlots fonctionnels suivant leur type

Dans l'illustration liée à la messagerie où le processus supporté est celui de l'usage d'une messagerie électronique, l'îlot fonctionnel *Sécurité de la messagerie* est de type flux pour les parcelles *Filtrer un message par rapport à l'objet du message*, *Filtrer le message par rapport à l'expéditeur du message* car toutes instances de ces parcelles ont une durée de vie dépendante du processus d'usage d'une messagerie électronique. L'îlot fonctionnel *sécurité de la messagerie* est de type stock pour sa parcelle *Créer les règles de filtrage de message*. La création de règles de filtrage des messages indésirables est en effet transverse aux différents usages d'un service de messagerie électronique. Le cycle de vie des instances de cette parcelle est donc indépendant de celui de l'usage du service. L'îlot fonctionnel *Réception d'un message* défini par la parcelle *Recevoir un message* est de type flux puisque cette parcelle a un cycle de vie dépendant de l'usage d'un service de messagerie

D'après l'Équation 3-1, la formulation de ce typage est la suivante :

$$t(\text{Sécurité de la messagerie}) = \{Flux, Stock\}$$

$$t(\text{Réception d'un message}) = \{Flux\}$$

3.3.2 Transformation des îlots du PLU fonctionnel contrainte par le typage

Certains îlots typés par les experts d'une zone fonctionnelle peuvent posséder des parcelles de type différent. Dans ce cas, il y a scission de l'îlot en autant d'îlots que de types distincts de ses parcelles. Cette sous-activité est la première partie de l'activité de conception automatisée des îlots et des voies fonctionnels entourée dans la **Figure 28**.

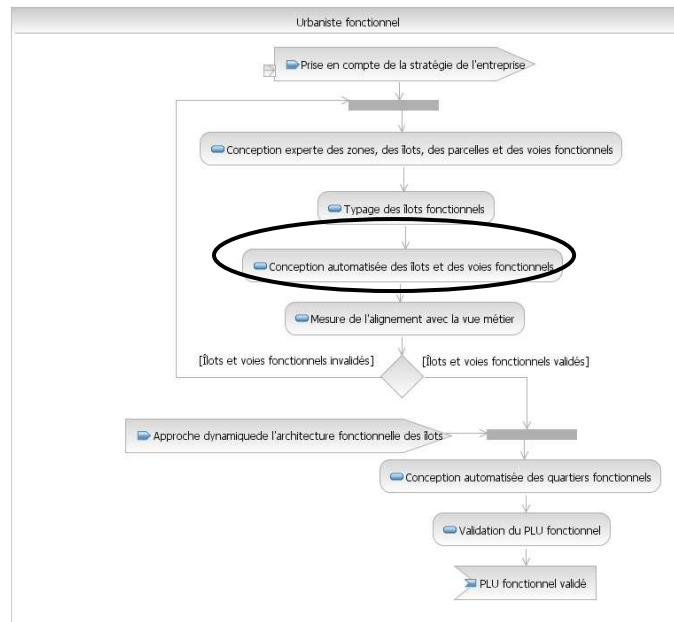


Figure 28 : Sous-activité de transformation des îlots fonctionnels par typage dans la démarche DA4EA

La fonction s définie ci-après représente la scission d'un îlot fonctionnel.

Définition : si T est l'ensemble des types d'îlots fonctionnels, si I_f est l'ensemble des îlots fonctionnels du PLU et si $P(I_f \times T)$ est l'ensemble des parties de $I_f \times T$, c'est-à-dire l'ensemble des sous-ensembles de $I_f \times T$, alors la scission s de l'îlot fonctionnel i en n îlots fonctionnels, dont i_1 de type t_1 et i_n de type t_n , peut être représentée de la manière suivante :

$$s : I_f \rightarrow P(I_f \times T)$$

$$i \mapsto \{(i_1, t_1), \dots, (i_n, t_n)\}; t(i_1) = \{t_1\}, \dots, t(i_n) = \{t_n\}$$

Équation 3-2

Dans la démarche DA4EA, il existe deux types d'îlots fonctionnels, soient le type flux et le type stock. Tous les îlots fonctionnels définis par la fonction s sont dépendants les uns des autres. Les parcelles d'un même îlot sont fortement couplées d'un point de vue fonctionnel. Ce couplage est représenté fonctionnellement par une voie entre les îlots issus de la scission. Le modèle des voies issues de la scission d'îlots fonctionnels est conforme à la règle de dépendance entre îlots fonctionnels suivant leur type défini dans la Règle d'urbanisme – Flux et stock du §3.3.1.

Dans l'illustration choisie de sécurité liée à la réception d'un message, si l'îlot fonctionnel *Sécurité de la messagerie* est à la fois du type flux et du type stock, alors cet îlot doit être scindé en deux îlots fonctionnels :

- *Sécurité de la messagerie – Filtrage de message* de type flux défini par les parcelles *Filtrer un message par rapport à l'objet du message* et *Filtrer le message par rapport à l'expéditeur du message*,
- *Sécurité de la messagerie – Gérer les règles de filtrage* de type stock défini par la parcelle *Créer les règles de filtrage de message*.

Ceci est équivalent, d'après l'Équation 3-2 à :

$$\Rightarrow s(\text{Sécurité de la messagerie}) = \left\{ \begin{array}{l} (\text{Sécurité de la messagerie filtrage de message, Flux}), \\ (\text{Sécurité de la messagerie gérer les règles de filtrage, Stock}) \end{array} \right\}$$

Le lien entre chaque îlot issu de la scission respecte la Règle d'urbanisme – Flux et stock. Le nouvel îlot fonctionnel *Sécurité de la messagerie – Filtrage de message* dépend donc de l'îlot fonctionnel *Sécurité de la messagerie – Gérer les règles de filtrage*. La dépendance de ces deux îlots fonctionnels est représentée **Figure 29**. Dans ce diagramme de classes UML, le type d'un îlot est inséré dans le stéréotype de la classe UML de l'îlot. Par exemple, le stéréotype de l'îlot fonctionnel *Sécurité de la messagerie – Filtrage de message* est "îlot fonctionnel – flux".

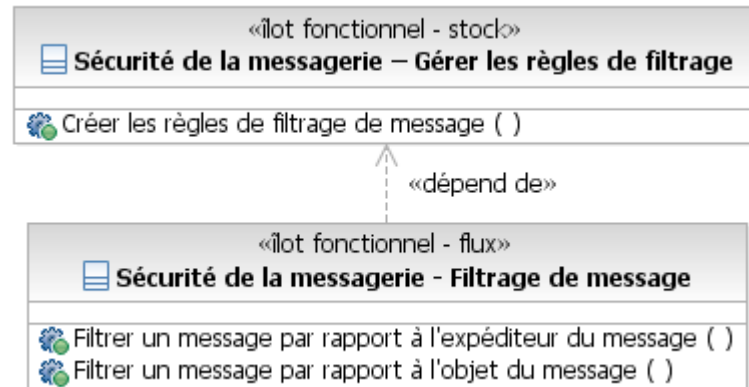


Figure 29 : Illustration de la scission d'un îlot fonctionnel respectant la contrainte par typage des îlots

3.3.3 Transformation des voies du PLU fonctionnel contrainte par le typage des îlots

La sous-activité de transformation des voies du PLU fonctionnel suite au typage des îlots fonctionnels est la seconde partie de l'activité de conception automatisée des îlots et des voies fonctionnels entourée dans la **Figure 30**.

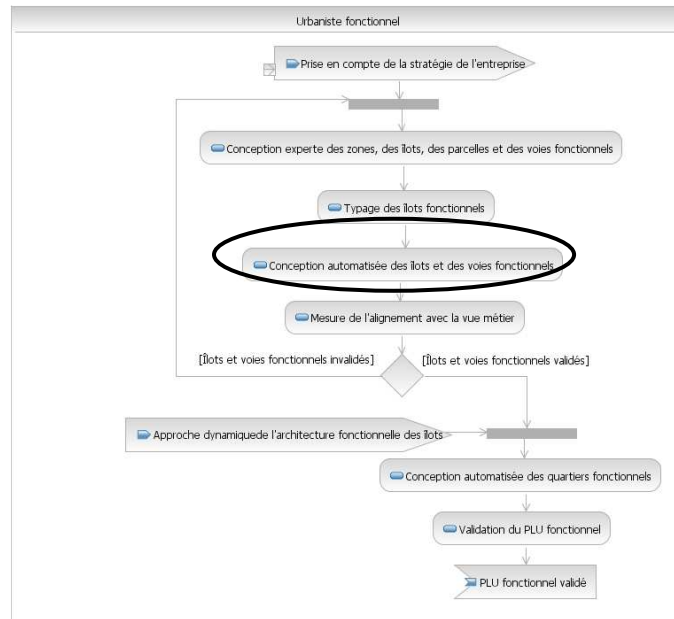


Figure 30 : Sous-activité de transformation des voies fonctionnelles par typage dans la démarche DA4EA

Lors d'une scission d'îlot fonctionnel, la règle d'urbanisme suivante permet de transformer les voies ayant pour source ou pour cible l'îlot scindé.

Règle d'urbanisme – Scission îlot multi-typé n°5

Lors de la scission d'un îlot fonctionnel :

- toutes dépendances ayant pour source l'îlot fonctionnel scindé a pour source les îlots fonctionnels issus de la scission ;
- toutes dépendances ayant pour cible l'îlot fonctionnel scindé a pour source les îlots fonctionnels issus de la scission.

Dans le diagramme de classes de la **Figure 31**, la dépendance avec l'îlot fonctionnel *Réception d'un message* est transformée en :

- une dépendance de l'îlot fonctionnel *Sécurité de la messagerie – Filtrage de message* de type flux vers l'îlot fonctionnel *Réception d'un message* de type flux qui satisfait la Règle d'urbanisme – Flux et stock.

La dépendance de l'îlot fonctionnel *Sécurité de la messagerie – Gérer les règles de filtrage* de type stock vers l'îlot fonctionnel *Réception d'un message* de type flux ne peut être créée conformément à la Règle d'urbanisme – Scission îlot multi-typé car elle ne respecte pas la Règle d'urbanisme – Flux et stock.

Il est intéressant de noter que l'absence de report de cette voie dans le modèle d'îlots fonctionnels a un sens métier. La gestion des règles de filtrage de message est en effet indépendante de la réception d'un message.

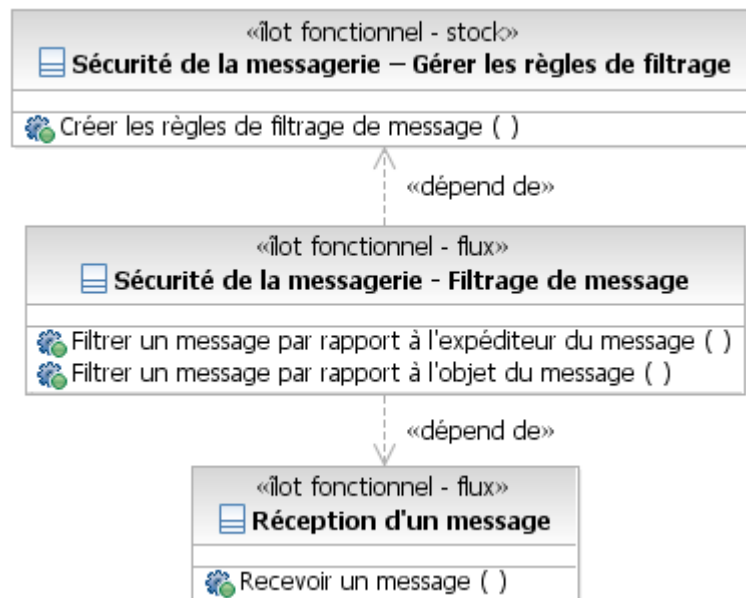


Figure 31 : Illustration des îlots et des voies fonctionnels respectant la contrainte par typage des îlots

Le diagramme de séquence UML de la **Figure 32** représente un scénario possible instanciant les dépendances entre les îlots issus de la scission de l'îlot fonctionnel *Sécurité de la messagerie*.

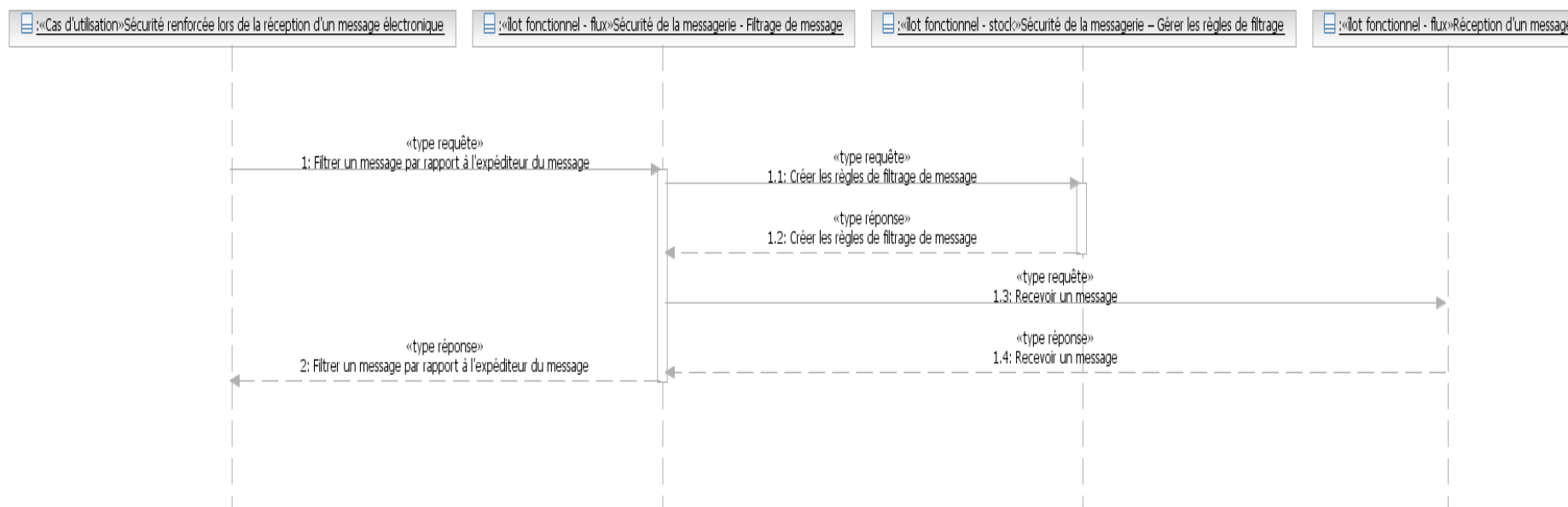


Figure 32 : Illustration d'un scénario instanciant des îlots respectant la contrainte par typage des îlots

Une interaction a pour source une instance de l'îlot fonctionnel *Sécurité de la messagerie – Filtrage de message* et pour cible une instance de l'îlot fonctionnel de *Sécurité de la messagerie – Gérer les règles de filtrage* puisque le filtrage d'un message dépend des règles de filtrage applicables. Une interaction a de plus pour source une instance de l'îlot fonctionnel *Sécurité de la messagerie – Filtrage de message* et pour cible une instance de l'îlot fonctionnel *Réception d'un message* pour exprimer la dépendance entre le filtrage d'un message et sa réception.

3.4 Alignement de la vue métier d'une entreprise et de la vue fonctionnelle d'un SI

L'architecture des îlots et des voies de la vue fonctionnelle étant stabilisée par le typage des îlots, un indicateur intéressant pour l'urbaniste fonctionnel est la cohérence de cette vue fonctionnelle par rapport à la vue métier. Cette cohérence reflète l'écart de la prise en compte de la stratégie de l'entreprise entre ces deux vues.

L'activité d'alignement de la vue fonctionnelle du SI avec la vue métier de l'entreprise ainsi que son évaluation est entourée dans la **Figure 33**.

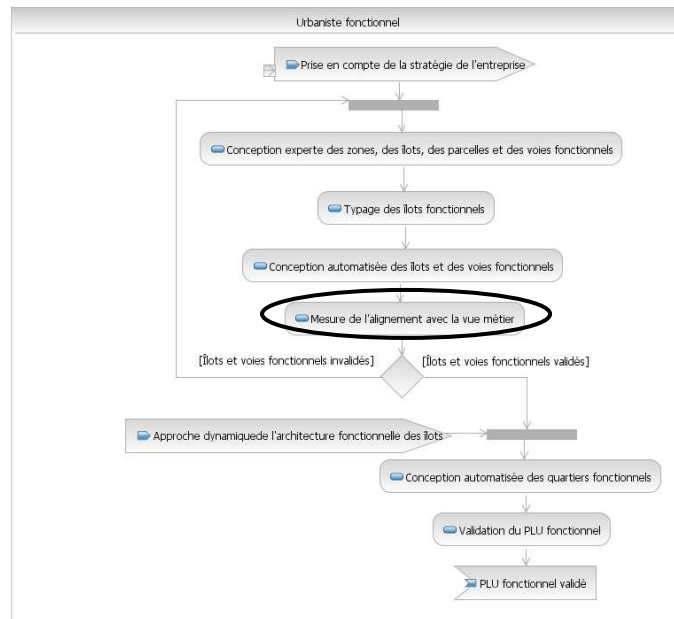


Figure 33 : Activité de mesure de l'alignement avec la vue métier dans la démarche DA4EA

3.4.1 Définition de l'alignement de la vue fonctionnelle avec la vue métier

L'alignement d'une vue fonctionnelle avec une vue métier nécessite de définir les critères d'alignement. L'intérêt du méta-modèle des vues du cadre de l'urbanisme est de proposer des concepts décrivant chaque vue. Le critère d'alignement consiste à

- associer des concepts de la vue métier et des concepts de la vue fonctionnelle décrits respectivement dans le §2.2.1.2 et dans le §2.2.1.3,
- vérifier leur cohérence.

De façon macroscopique, deux grandes associations entre concepts de la vue métier et de la vue fonctionnelle peuvent être étudiées :

- l'association entre les données métier de la vue métier et les données logiques de la vue fonctionnelle,
- l'association entre les activités métier de la vue métier et les parcelles fonctionnelles de la vue fonctionnelle.

Le choix est ici d'associer les activités métier et les parcelles fonctionnelles. L'alignement entre les données métier et les données logiques se déduit en effet de l'alignement entre les activités métier et les parcelles fonctionnelles puisque, par définition,

- une donnée métier est produite par une activité métier,
- une donnée logique est produite par une ou plusieurs parcelles fonctionnelles d'un même îlot.

L'association entre une relation de succession d'activités métier et une relation de dépendance de parcelles fonctionnelles définies par la Sémantique UML – Approche dynamique et diagramme de séquence (cf. §2.1.5) est aussi ciblée par cet alignement. Ces deux types de relation caractérisent l'approche dynamique.

Définition :

- deux activités métier sont en relation de succession si l'une d'entre elles succède temporellement à l'autre dans un diagramme d'activités UML représentant une procédure métier (par exemple dans la **Figure 34**, la relation de succession issue de l'activité métier *Réception d'un message* vers l'activité métier *Filtrage appliqué au message*) ;
- deux parcelles fonctionnelles sont en relation de dépendance si toutes deux sont exclusivement du type requête ou du type réponse et si l'une d'entre elles succède à l'autre dans un diagramme de séquence UML représentant un scénario. Une parcelle P_1 succède à une parcelle P_2 si la ligne de vie cible de P_2 est la ligne de vie à l'origine de P_1 (par exemple dans la **Figure 32**, la relation de dépendance de la parcelle fonctionnelle *Filtrer un message par rapport à l'expéditeur du message* vers la parcelle fonctionnelle *Créer les règles de filtrage de message* ou la relation de dépendance de la parcelle fonctionnelle *Filtrer un message par rapport à l'expéditeur du message* vers la parcelle fonctionnelle *Recevoir un message*).

Définition : l'alignement de la vue fonctionnelle avec la vue métier est défini de la façon suivante,

- une parcelle fonctionnelle est alignée avec la vue métier
 - o si la parcelle fonctionnelle a un sens métier commun avec au moins une activité de la vue métier,
 - o et si chaque activité métier alignée avec la parcelle a au moins une relation de succession avec une des autres activités métier alignées avec la parcelle ;
- une relation de dépendance entre deux parcelles fonctionnelles P_1 et P_2 , telle que P_1 dépend de P_2 , est alignée avec la vue métier s'il existe dans la vue métier,
 - o au moins une activité métier A_1 alignée avec P_1 ,
 - o au moins une activité métier A_2 alignée avec P_2 ,
 - o telles que A_1 succède à A_2 .

L'illustration proposée cible l'alignement des parcelles fonctionnelles et de leurs relations de dépendance décrites respectivement dans la **Figure 31** et dans la **Figure 32** avec les activités métier et leurs relations de succession décrites dans la **Figure 34**.

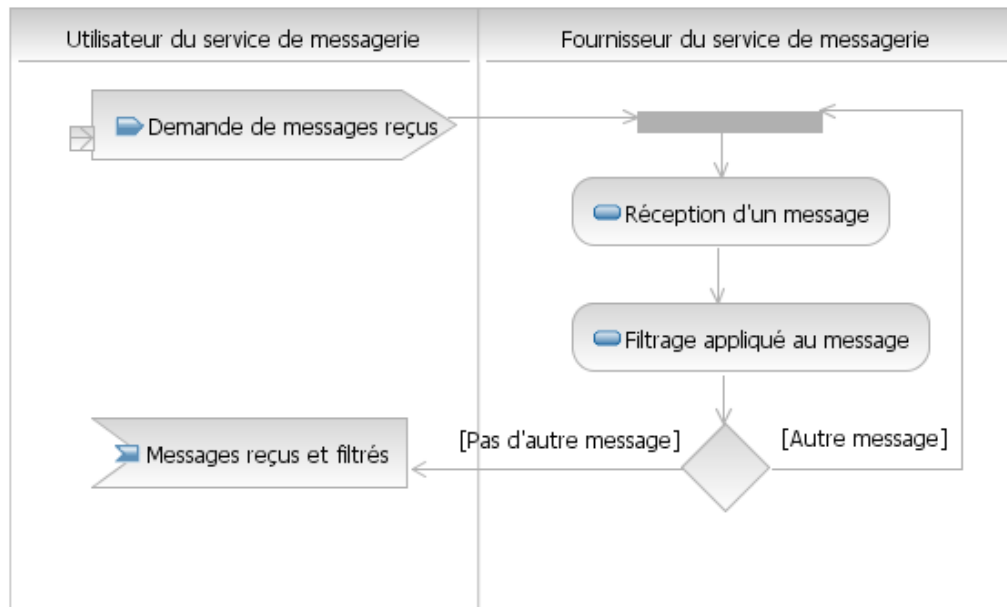


Figure 34 : Illustration d'une vue métier

L'alignement est celui :

- de la parcelle fonctionnelle *Recevoir un message*
 - o avec l'activité métier *Réception d'un message*,
- des parcelles *Filtrer un message par rapport à l'expéditeur du message* et *Filtrer un message par rapport à l'objet du message*
 - o avec l'activité métier *Filtrage appliqué au message*,
- de la relation de dépendance de la parcelle fonctionnelle *Filtrer un message par rapport à l'expéditeur du message* vis-à-vis de la parcelle fonctionnelle *Recevoir un message*
 - o avec la relation de succession de l'activité métier *Réception d'un message* vers l'activité métier *Filtrage appliqué au message*

La parcelle fonctionnelle *Créer les règles de filtrage de message* et la relation de dépendance de la parcelle fonctionnelle *Filtrer un message par rapport à l'expéditeur du message* vis-à-vis de la parcelle fonctionnelle *Créer les règles de filtrage de message* ne sont pas alignées avec la vue métier. Aucune activité du diagramme de la **Figure 34** n'a en effet de sens métier commun avec la parcelle de gestion des règles de filtrage.

3.4.2 Axiomatisation de l'alignement de la vue fonctionnelle avec la vue métier

Le comportement attendu de l'alignement des vues métier et fonctionnelle a pour paramètres les concepts définissant cet alignement (cf. §3.4.1). Une axiomatisation permet de décrire ce comportement attendu [Shepperd, 1993]. La mesure choisie pour évaluer cet alignement doit donc vérifier des axiomes.

L'alignement des vues métier et fonctionnelle est soit l'alignement de la vue fonctionnelle avec la vue métier soit l'alignement de la vue métier avec la vue fonctionnelle. Pour l'urbaniste fonctionnel, la conception de la vue fonctionnelle nécessite d'étudier le premier alignement.

Les quatre axiomes suivants décrivent l'Alignement de la vue Fonctionnelle par rapport à la vue Métier (AFM). Ils ciblent les concepts de la vue fonctionnelle définissant l'alignement.

AFM1 – Ajout d'une nouvelle parcelle dans la vue fonctionnelle : l'alignement résultant de l'ajout d'une nouvelle parcelle dans la vue fonctionnelle est, par rapport à l'alignement précédent,

- moins bon ou identique si la parcelle n'a pas de sens métier commun avec au moins une activité de la vue métier,
- meilleur ou identique si la parcelle a un sens métier commun avec au moins une activité de la vue métier.

AFM2 – Ajout d'une relation de dépendance entre deux parcelles fonctionnelles : l'alignement résultant de l'ajout d'une relation de dépendance entre deux parcelles fonctionnelles est, par rapport à l'alignement précédent,

- moins bon ou identique s'il n'existe pas de relation de succession entre deux activités de la vue métier alignée avec la relation de dépendance ajoutée,
- meilleur ou identique s'il existe au moins une relation de succession entre deux activités de la vue métier alignée avec la relation de dépendance ajoutée

AFM3 – Suppression d'une parcelle de la vue fonctionnelle : l'alignement résultant de la suppression d'un îlot de la vue fonctionnelle est, par rapport à l'alignement précédent,

- moins bon ou identique si la parcelle a un sens métier commun avec au moins une activité de la vue métier,
- meilleur ou identique si la parcelle n'a pas de sens métier commun avec au moins une activité de la vue métier.

AFM4 – Suppression d'une relation de dépendance entre deux parcelles fonctionnelles : l'alignement résultant de la suppression d'une relation de dépendance entre deux parcelles fonctionnelles est, par rapport à l'alignement précédent,

- moins bon ou identique s'il existe au moins une relation de succession entre deux activités de la vue métier alignée avec la relation de dépendance supprimée,
- meilleur ou identique s'il n'existe pas de relation de succession entre deux activités de la vue métier alignée avec la relation de dépendance supprimée.

3.4.3 Mesure de l'alignement de la vue fonctionnelle avec la vue métier

La mesure de l'alignement est un outil mis à la disposition de l'urbaniste fonctionnel afin de vérifier la cohérence de sa vue fonctionnelle par rapport à la vue métier de l'entreprise. Les axiomes à vérifier sont les axiomes AFM1, AFM2, AFM3, AFM4 décrits dans le §3.4.2. La mesure proposée est le produit du pourcentage de parcelles alignées par le pourcentage de relations de dépendance entre parcelles alignées.

Définition : La mesure MAFM de l'alignement de la vue fonctionnelle VF par rapport à la vue métier conforme à l'axiomatisation est telle que :

$$MAFM(VF) = \left(\frac{N_p(VF) - N_{nap}(VF)}{N_p(VF)} \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right)$$

Équation 3-3

avec pour paramètres,

- le nombre $N_p(VF)$ de parcelles de la vue fonctionnelle VF,
- le nombre $N_{nap}(VF)$ de parcelles de la vue fonctionnelle VF non alignées avec des activités de la vue métier,
- le nombre $N_r(VF)$ de relations de dépendance entre parcelles de la vue fonctionnelle VF,
- le nombre $N_{nar}(VF)$ de relations de dépendance entre parcelles de la vue fonctionnelle VF non alignées avec des relations de succession entre activités de la vue métier.

La mesure MAFM prend toutes ses valeurs dans l'intervalle $[0, 1]$. MAFM vaut 0 lorsqu'aucune parcelle de la vue fonctionnelle ou aucune relation de dépendance entre parcelles n'est alignée avec la vue métier. A l'opposé, MAFM vaut 1 lorsque l'alignement est parfait, c'est-à-dire lorsque toutes les parcelles et leurs relations de dépendance sont alignées avec la vue métier.

Proposition : la mesure MAFM vérifie les quatre axiomes AFM1, AFM2, AFM3, AFM4 de l'alignement fonctionnel par rapport à la vue métier.

Démonstration :

- AFM1 : Soit P une parcelle fonctionnelle ajoutée à la vue fonctionnelle VF, d'après l'Équation 3-3,

$$MAFM(VF \cup \{P\}) = \left(\frac{N_p(VF \cup \{P\}) - N_{nap}(VF \cup \{P\})}{N_p(VF \cup \{P\})} \right) * \left(\frac{N_r(VF \cup \{P\}) - N_{nar}(VF \cup \{P\})}{N_r(VF \cup \{P\})} \right) =$$

$$\left(\frac{(N_p(VF) + 1) - (N_{nap}(VF) + N_{nap}(\{P\}))}{N_p(VF) + 1} \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right)$$

puisque le nombre de relations de dépendance entre parcelles n'est pas modifié.

$$\Rightarrow MAFM(VF \cup \{P\}) - MAFM(VF) =$$

$$\left(\left(1 - \frac{N_{nap}(VF) + N_{nap}(\{P\})}{N_p(VF) + 1} \right) - \left(1 - \frac{N_{nap}(VF)}{N_p(VF)} \right) \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right) =$$

$$\left(\frac{N_{nap}(VF)}{N_p(VF)} - \frac{N_{nap}(VF) + N_{nap}(\{P\})}{N_p(VF) + 1} \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right)$$

- o Si la parcelle n'a pas de sens métier commun avec au moins une activité de la vue métier alors $N_{nap}(\{P\}) = 1$

$$\Rightarrow MAFM(VF \cup \{P\}) - MAFM(VF) =$$

$$\left(\frac{N_{nap}(VF)}{N_p(VF)} - \frac{N_{nap}(VF) + 1}{N_p(VF) + 1} \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right)$$

Or,

$$\frac{N_{nap}(VF)}{N_p(VF)} - \frac{N_{nap}(VF) + 1}{N_p(VF) + 1} = \frac{N_{nap}(VF) - N_p(VF)}{N_p(VF) * (N_p(VF) + 1)} \leq 0$$

$$\Rightarrow MAFM(VF \cup \{P\}) \leq MAFM(VF)$$

puisque le nombre de parcelles non alignées est inférieur ou égal au nombre total de parcelles.

Il y a égalité des mesures d'alignement lorsque

$$N_p(VF) = N_{nap}(VF)$$

c'est-à-dire, lorsqu'aucune parcelle de la vue fonctionnelle VF est alignée avec la vue métier.

- Si la parcelle a un sens métier commun avec au moins une activité de la vue métier alors $N_{nap}(\{P\}) = 0$

$$\Rightarrow MAFM(VF \cup \{P\}) - MAFM(VF) =$$

$$\left(\frac{N_{nap}(VF)}{N_p(VF)} - \frac{N_{nap}(VF)}{N_p(VF)+1} \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right)$$

$$\text{Or, } \frac{N_{nap}(VF)}{N_p(VF)} - \frac{N_{nap}(VF)}{N_p(VF)+1} = \frac{N_{nap}(VF)}{N_p(VF) * (N_p(VF)+1)} \geq 0$$

$$\Rightarrow MAFM(VF \cup \{P\}) \geq MAFM(VF)$$

Il y a égalité des mesures d'alignement lorsque $N_{nap}(VF) = 0$, c'est-à-dire lorsque toutes les parcelles de la vue fonctionnelle VF sont alignées avec la vue métier.

- AFM2 est vérifié de la même façon par la mesure MAFM puisque $N_r(VF)$ et $N_p(VF)$ d'un côté, $N_{nar}(VF)$ et $N_{nap}(VF)$ ont un rôle symétrique dans l'Équation 3-3.
- AFM3 : Soit P une parcelle fonctionnelle supprimée de la vue fonctionnelle VF, d'après l'Équation 3-3,

$$MAFM(VF - \{P\}) = \left(\frac{N_p(VF - \{P\}) - N_{nap}(VF - \{P\})}{N_p(VF - \{P\})} \right) * \left(\frac{N_r(VF - \{P\}) - N_{nar}(VF - \{P\})}{N_r(VF - \{P\})} \right) =$$

$$\left(\frac{(N_p(VF) - 1) - (N_{nap}(VF) - N_{nap}(\{P\}))}{N_p(VF) - 1} \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right)$$

puisque le nombre de relations de dépendance entre parcelles n'est pas modifié.

$$\Rightarrow MAFM(VF - \{P\}) - MAFM(VF) =$$

$$\left(\left(1 - \frac{N_{nap}(VF) - N_{nap}(\{P\})}{N_p(VF) - 1} \right) - \left(1 - \frac{N_{nap}(VF)}{N_p(VF)} \right) \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right) =$$

$$\left(\frac{N_{nap}(VF)}{N_p(VF)} - \frac{N_{nap}(VF) - N_{nap}(\{P\})}{N_p(VF) - 1} \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right)$$

- Si la parcelle n'a pas de sens métier commun avec au moins une activité de la vue métier alors $N_{nap}(\{P\}) = 1$

$$\Rightarrow MAFM(VF - \{P\}) - MAFM(VF) =$$

$$\left(\frac{N_{nap}(VF)}{N_p(VF)} - \frac{N_{nap}(VF) - 1}{N_p(VF) - 1} \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right)$$

$$\text{Or } \frac{N_{nap}(VF)}{N_p(VF)} - \frac{N_{nap}(VF) - 1}{N_p(VF) - 1} = \frac{N_p(VF) - N_{nap}(VF)}{N_p(VF) * (N_p(VF) - 1)} \geq 0$$

$$\Rightarrow MAFM(VF - \{P\}) \leq MAFM(VF)$$

puisque le nombre de parcelles non alignées est inférieur ou égal au nombre total de parcelles.

Il y a égalité des mesures d'alignement lorsque $N_p(VF) = N_{nap}(VF)$, c'est-à-dire lorsqu'aucune parcelle de la vue fonctionnelle VF est alignée avec la vue métier.

- Si la parcelle a un sens métier commun avec au moins une activité de la vue métier alors $N_{nap}(\{P\}) = 0$

$$\Rightarrow MAFM(VF - \{P\}) - MAFM(VF) =$$

$$\left(\frac{N_{nap}(VF)}{N_p(VF)} - \frac{N_{nap}(VF)}{N_p(VF) - 1} \right) * \left(\frac{N_r(VF) - N_{nar}(VF)}{N_r(VF)} \right)$$

$$\text{Or } \frac{N_{nap}(VF)}{N_p(VF)} - \frac{N_{nap}(VF)}{N_p(VF) - 1} = - \frac{N_{nap}(VF)}{N_p(VF) * (N_p(VF) - 1)} \leq 0$$

$$\Rightarrow MAFM(VF - \{P\}) \leq MAFM(VF)$$

Il y a égalité des mesures d'alignement lorsque $N_{nap}(VF) = 0$, c'est-à-dire lorsque toutes les parcelles de la vue fonctionnelle VF sont alignées avec la vue métier.

- AFM4 est vérifié de la même façon par la mesure MAFM puisque $N_r(VF)$ et $N_p(VF)$ d'un côté, $N_{nar}(VF)$ et $N_{nap}(VF)$ ont un rôle symétrique dans l'Équation 3-3.

Dans l'illustration choisie de l'alignement de la vue fonctionnelle VF, représentée respectivement pour les parcelles dans la **Figure 31** et pour les relations de dépendance entre parcelles dans la **Figure 32**, avec la vue métier, représentée **Figure 34** :

- $N_p(VF) = 4$,
- $N_{nap}(VF) = 1$ puisque la parcelle *Créer les règles de filtrage de message* de la **Figure 31** ne peut être alignée avec la vue métier,
- $N_r(VF) = 2$,
- $N_{nar}(VF) = 1$ puisque la relation de dépendance de la parcelle *Filtrer un message par rapport à l'expéditeur du message* vers la parcelle *Créer les règles de filtrage de message* de la **Figure 32** ne peut être alignée avec la vue métier.

La mesure MAFM de l'alignement de la vue fonctionnelle VF par rapport à la vue métier conforme à l'axiomatisation est alors :

$$MAFM(VF) = \left(\frac{4-1}{4} \right) * \left(\frac{2-1}{2} \right) = \frac{3}{8}$$

75% des parcelles et 50% des relations de dépendance entre parcelles sont en effet alignées avec la vue métier. La mesure pourrait être améliorée par l'urbaniste fonctionnel en estimant que la gestion des règles de filtrage d'un message ne fait pas partie de la vue fonctionnelle du SI. L'urbaniste doit alors vérifier ce choix avec la stratégie de l'entreprise.

3.5 Conception automatisée des quartiers du PLU fonctionnel

Les îlots et les voies d'une même zone de la vue fonctionnelle ainsi stabilisés par le typage des parcelles de chaque îlot et par la mesure de l'alignement de la vue fonctionnelle avec la vue métier peuvent alors être regroupés dans des quartiers.

Chaque urbaniste fonctionnel conçoit les quartiers fonctionnels de la zone dont il est responsable lors de l'activité entourée dans la **Figure 35**.

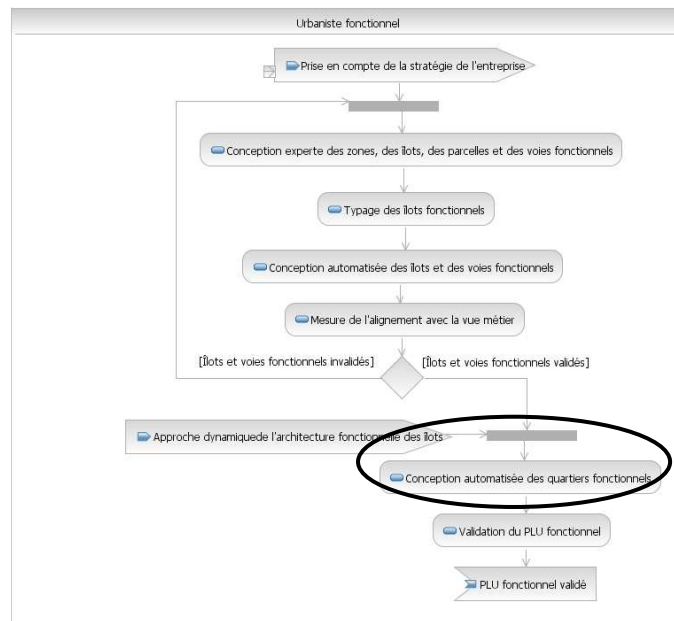


Figure 35 : Activité de conception automatisée des quartiers dans la démarche DA4EA

La méthode de conception des quartiers proposée dans ce paragraphe a pour objectif d'assister l'urbaniste fonctionnel par une méthode automatisée fondée sur une approche dynamique de la conception des îlots de la vue fonctionnelle.

3.5.1 Approche dynamique de la conception des quartiers

L'activité de conception des quartiers fonctionnels a pour objectif de regrouper dans un quartier des îlots fonctionnels dont le couplage est significatif. La mesure du couplage de deux îlots fonctionnels nécessite une approche dynamique. Elle s'appuie en effet sur la transformation d'un ensemble de séquences d'instances d'îlots fonctionnels spécifiques à la zone étudiée en un ensemble de regroupements disjoints d'îlots pouvant former des quartiers. Les séquences d'instances d'îlots fonctionnels représentent des scénarios de cas d'utilisation propres à la zone étudiée par l'urbaniste fonctionnel.

Les interactions entre les îlots fonctionnels instanciés dans ces séquences déterminent un couplage entre les îlots fonctionnels. Cette déduction est due à la conformité des interactions entre instances d'îlots fonctionnels avec les voies fonctionnelles entre ces îlots (cf. Règle d'urbanisme – Interactions et voies du §3.2). Le principe est de lier le couplage entre deux îlots avec le nombre d'interactions entre des instances des ces îlots durant les scénarios de cas d'utilisation propres à la zone.

Le couplage entre îlots fonctionnels est représenté sur le diagramme de séquence UML de la **Figure 36**. Chaque interaction de type requête entre instances d'îlots fonctionnels est un paramètre du couplage entre les îlots instanciés.

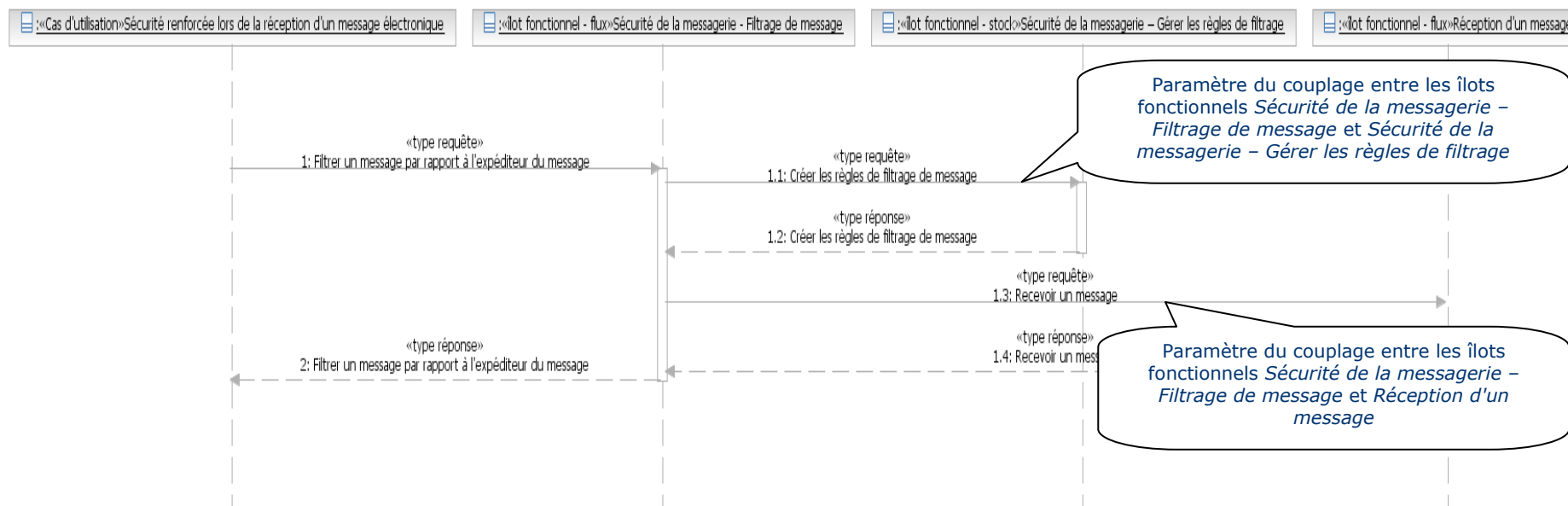


Figure 36 : Illustration du couplage entre îlots fonctionnels à partir d'une approche dynamique de la vue fonctionnelle des îlots

Ce scénario augmente donc le couplage :

- entre les îlots fonctionnels *Sécurité de la messagerie - Filtrage de message* et *Sécurité de la messagerie - Gérer les règles de filtrage*,
- entre les îlots fonctionnels *Sécurité de la messagerie - Filtrage de message* et *Réception d'un message*.

3.5.2 Pertinence d'un regroupement d'îlots fonctionnels dans une même entité fonctionnelle

Afin de mesurer le couplage d'un ensemble d'îlots fonctionnels, un indicateur de pertinence de cet ensemble est proposé [INPI – 05929] [INPI – 06471].

Définition : Une entité fonctionnelle est un regroupement d'îlots fonctionnels. Un quartier fonctionnel est une entité fonctionnelle particulière où les îlots regroupés ont un couplage significatif.

Afin de proposer une mesure de la pertinence d'un regroupement d'îlots fonctionnels, des axiomes du comportement de cette pertinence du regroupement d'îlots sont définis. Les concepts à prendre en compte sont ceux d'îlot fonctionnel et de relation de dépendance entre îlots fonctionnels.

Définition : Deux îlots fonctionnels sont en relation de dépendance s'il existe une interaction de type requête de l'un vers l'autre dans un diagramme de séquence UML. L'îlot à l'origine de l'interaction dépend de l'îlot à la cible de cette interaction.

C'est le cas dans la **Figure 32** de la relation de dépendance de l'îlot fonctionnel *Sécurité de la messagerie – Filtrage de message* vis-à-vis de l'îlot fonctionnel *Sécurité de la messagerie – Gérer les règles de filtrage* ou de la relation de dépendance de l'îlot fonctionnel *Sécurité de la messagerie – Filtrage de message* vis-à-vis de l'îlot fonctionnel *Réception d'un message*.

Les axiomes suivants caractérisent la Pertinence du Regroupement d'îlots fonctionnels (PRI) dans une même entité fonctionnelle. Ils sont fondés sur le fait que la pertinence est assimilée à la longueur moyenne des séquences internes à l'entité fonctionnelle regroupant des îlots fonctionnels. Une séquence interne à une entité fonctionnelle est une séquence, sans interruption, de relations de dépendance entre îlots de l'entité fonctionnelle.

PRI1 – Ajout d'un îlot fonctionnel dans l'entité fonctionnelle : la pertinence d'une entité fonctionnelle résultante de l'ajout d'un îlot fonctionnel dans l'entité fonctionnelle est, par rapport à la pertinence précédente,

- identique s'il n'y a pas de relation de dépendance entre l'îlot ajouté et les autres îlots de l'entité fonctionnelle,
- meilleure s'il y a au moins une relation de dépendance entre l'îlot ajouté et les autres îlots de l'entité fonctionnelle.

PRI2 – Ajout d'une nouvelle relation de dépendance entre îlots fonctionnels : la pertinence d'une entité fonctionnelle résultante de l'ajout d'une nouvelle relation de dépendance entre îlots fonctionnels est, par rapport à la pertinence précédente,

- identique si la relation de dépendance est entre deux îlots fonctionnels dont au moins un n'appartient pas à l'entité fonctionnelle,
- meilleure si la relation de dépendance est entre deux îlots fonctionnels appartenant à l'entité fonctionnelle.

PRI3 – Suppression d'un îlot fonctionnel de l'entité fonctionnelle : la pertinence d'une entité fonctionnelle résultante de la suppression d'un îlot fonctionnel de l'entité fonctionnelle est, par rapport à la pertinence précédente,

- moins bonne s'il y a au moins une relation de dépendance entre l'îlot supprimé et un autre îlot de l'entité fonctionnelle,
- identique s'il n'y a pas de relation de dépendance entre l'îlot supprimé et un autre îlot de l'entité fonctionnelle.

PRI4 – Suppression d'une relation de dépendance entre îlots fonctionnels : la pertinence d'une entité fonctionnelle résultante de la suppression d'une nouvelle relation de dépendance entre îlots fonctionnels est, par rapport à la pertinence précédente,

- moins bonne si la relation de dépendance est entre deux îlots fonctionnels appartenant à l'entité fonctionnelle.
- identique si la relation de dépendance est entre deux îlots fonctionnels dont au moins un n'appartient pas à l'entité fonctionnelle.

La mesure de la pertinence d'un regroupement d'îlots fonctionnels dans une entité fonctionnelle E est un outil mis à la disposition de l'urbaniste fonctionnel afin de l'assister dans la conception des quartiers fonctionnels d'une zone du PLU fonctionnel. Les axiomes à vérifier sont les axiomes PRI1, PRI2, PRI3, PRI4. La mesure proposée est la longueur moyenne, en termes de relations de dépendance entre îlots fonctionnels, des séquences internes à l'entité fonctionnelle E.

Définition : La mesure MPRI de la pertinence d'un regroupement d'îlots fonctionnels dans une même entité fonctionnelle E à partir d'un ensemble Se de séquences d'instances d'îlot fonctionnel réalisées par un ensemble Sc de scénarios est telle que :

$$MPRI(E, Sc, Se) = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se)}{\sum_{i \geq 1} N_i(E, Se)}$$

Équation 3-4

avec :

- $N_i(E, Se)$, le nombre de séquences de i instances d'îlots fonctionnels appartenant à E extraites de Se.

MPRI vaut 0 s'il n'existe aucune séquence de Se contenant strictement plus d'une instance d'îlot fonctionnel de E.

Remarque : l'ensemble Se et l'ensemble Sc sont différents car l'instance du cas d'utilisation représentant le scénario n'est pas une instance d'îlot fonctionnel.

Les séquences d'instances d'îlots fonctionnels de Se prises en compte dans le calcul de l'indicateur de pertinence d'une entité fonctionnelle sont les séquences internes à l'entité fonctionnelle de longueur maximale.

La notion de "séquence interne à l'entité fonctionnelle de longueur maximale" signifie que si $E = \{C1, C2, C3\}$ et s'il existe dans une séquence de Se, l'enchaînement d'interactions $C1 \rightarrow C2 \rightarrow C3$, alors les enchaînements d'interactions incluses $C1 \rightarrow C2$ et $C2 \rightarrow C3$ ne seraient pas à prendre en compte pour le calcul de $MPRI(E, Sc, Se)$.

Si $l(s)$ est la longueur d'une séquence s de Se, en termes de relations de dépendances, la mesure MPRI prend toutes ses valeurs dans l'intervalle $\left[0, \frac{\sum_{s \in Se} l(s)}{Card(Se)}\right]$.

MPRI vaut 0 lorsqu'aucune relation de dépendance entre îlots fonctionnels n'existe dans

Se. MPRI vaut $\frac{\sum_{s \in Se} l(s)}{Card(Se)}$ lorsque E contient tous les îlots fonctionnels instanciés dans les séquences de Se.

Proposition : la mesure MPRI vérifie les quatre axiomes PRI1, PRI2, PRI3, PRI4 de la pertinence d'un regroupement d'îlots fonctionnels dans une même entité fonctionnelle.

Démonstration :

- PRI1 : Soit I un îlot fonctionnel ajouté dans l'entité fonctionnelle E, d'après l'Équation 3-4,

$$MPRI(E \cup \{I\}, Sc, Se) = \frac{\sum_{i \geq 2} (i-1) * N_i(E \cup \{I\}, Se)}{\sum_{i \geq 1} N_i(E \cup \{I\}, Se)}$$

- o Si l'îlot ajouté n'a pas de relation de dépendance avec les autres îlots de E, alors

$$\forall i \geq 1, N_i(E \cup \{I\}, Se) = N_i(E, Se)$$

$$\Rightarrow MPRI(E \cup \{I\}, Sc, Se) = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se)}{\sum_{i \geq 1} N_i(E, Se)} = MPRI(E, Sc, Se)$$

- o Si l'îlot ajouté a une relation de dépendance avec un autre îlot de E et si cette relation de dépendance est intégrée à une séquence de j îlots fonctionnels de Se, alors

$$\forall i \geq 1, i \neq j, i \neq j+1 \Rightarrow N_i(E \cup \{I\}, Se) = N_i(E, Se)$$

avec

$$N_j(E \cup \{I\}, Se) = N_j(E, Se) - 1$$

$$\text{et } N_{j+1}(E \cup \{I\}, Se) = N_{j+1}(E, Se) + 1$$

puisque'une séquence de j îlots est remplacée par une séquence de j + 1 îlots.

$$\Rightarrow MPRI(E \cup \{I\}, Sc, Se) =$$

$$\frac{\sum_{i \geq 2, i \neq j, i \neq j+1} (i-1) * N_i(E, Se) + ((j-1) * (N_j(E, Se) - 1)) + (j * (N_{j+1}(E, Se) + 1))}{\sum_{i \geq 1, i \neq j, i \neq j+1} N_i(E, Se) + (N_j(E, Se) - 1) + (N_{j+1}(E, Se) + 1)} =$$

$$\frac{\sum_{i \geq 2} (i-1) * N_i(E, Se) + ((j-1) * (-1)) + (j * (+1))}{\sum_{i \geq 1} N_i(E, Se)} = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se) + 1}{\sum_{i \geq 1} N_i(E, Se)}$$

$$\text{Or, } \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se) + 1}{\sum_{i \geq 1} N_i(E, Se)} > \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se)}{\sum_{i \geq 1} N_i(E, Se)}$$

$$\Rightarrow MPRI(E \cup \{I\}, Sc, Se) > MPRI(E, Sc, Se)$$

Conformément à la démonstration pour une séquence, l'inégalité est démontrée par récurrence de façon immédiate pour l'intégration de la relation de dépendance dans n séquences de Se.

- PRI2 : Soit R une relation de dépendance entre îlots fonctionnels ajouté à l'ensemble des séquences Se, d'après l'Équation 3-4,

$$MPRI(E, Sc, Se \cup \{R\}) = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se \cup \{R\})}{\sum_{i \geq 1} N_i(E, Se \cup \{R\})}$$

La démonstration est identique à celle pour l'axiome PRI1 puisque :

- o l'hypothèse que la relation de dépendance ajoutée ne soit pas interne à l'entité fonctionnelle est équivalente au cas où un îlot n'ayant pas de relation de dépendance avec un autre îlot de l'entité fonctionnelle est ajouté à celle-ci,
- o l'hypothèse que la relation de dépendance ajoutée soit interne à l'entité fonctionnelle est équivalente au cas où un îlot ayant une relation de dépendance avec un autre îlot de l'entité fonctionnelle est ajouté à celle-ci.
- PRI3 : Soit I un îlot fonctionnel supprimé de l'entité fonctionnelle E, d'après l'Équation 3-4,

$$MPRI(E - \{I\}, Sc, Se) = \frac{\sum_{i \geq 2} (i-1) * N_i(E - \{I\}, Se)}{\sum_{i \geq 1} N_i(E - \{I\}, Se)}$$

- o Si l'îlot supprimé a une relation de dépendance avec un autre îlot de E et si cette relation de dépendance est intégrée à une séquence de j îlots fonctionnels de Se, alors

$$\forall i \geq 1, i \neq j-1, i \neq j \Rightarrow N_i(E \cup \{I\}, Se) = N_i(E, Se)$$

$$\text{avec } N_{j-1}(E \cup \{I\}, Se) = N_{j-1}(E, Se) + 1 \text{ et}$$

$$N_j(E \cup \{I\}, Se) = N_j(E, Se) - 1$$

puisque une séquence de j îlots est remplacée par une séquence de j - 1 îlots.

$$\Rightarrow MPRI(E - \{I\}, Sc, Se) =$$

$$\frac{\sum_{i \geq 2, i \neq j-1, i \neq j} (i-1) * N_i(E, Se) + ((j-2) * (N_{j-1}(E, Se) + 1)) + ((j-1) * (N_j(E, Se) - 1))}{\sum_{i \geq 1, i \neq j-1, i \neq j} N_i(E, Se) + (N_{j-1}(E, Se) + 1) + (N_j(E, Se) - 1)} =$$

$$\frac{\sum_{i \geq 2} (i-1) * N_i(E, Se) + ((j-2) * (+1)) + ((j-1) * (-1))}{\sum_{i \geq 1} N_i(E, Se)} = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se) - 1}{\sum_{i \geq 1} N_i(E, Se)}$$

$$\text{Or, } \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se) - 1}{\sum_{i \geq 1} N_i(E, Se)} < \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se)}{\sum_{i \geq 1} N_i(E, Se)}$$

$$\Rightarrow MPRI(E \cup \{I\}, Sc, Se) < MPRI(E, Sc, Se)$$

Conformément à la démonstration pour une séquence, l'inégalité est démontrée par récurrence de façon immédiate pour l'intégration de la relation de dépendance dans n séquences de Se.

- o Si l'îlot ajouté n'a pas de relation de dépendance avec les autres îlots de E, alors

$$\forall i \geq 1, N_i(E \cup \{I\}, Se) = N_i(E, Se)$$

$$\Rightarrow MPRI(E \cup \{I\}, Sc, Se) = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se)}{\sum_{i \geq 1} N_i(E, Se)} = MPRI(E, Sc, Se)$$

- PRI4 : Soit R une relation de dépendance entre îlots fonctionnels ajouté à l'ensemble des séquences Se, d'après l'Équation 3-4,

$$MPRI(E, Sc, Se - \{R\}) = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se - \{R\})}{\sum_{i \geq 1} N_i(E, Se - \{R\})}$$

La démonstration est identique à celle pour l'axiome PRI3 puisque :

- l'hypothèse que la relation de dépendance supprimée soit interne à l'entité fonctionnelle est équivalente au cas où un îlot ayant une relation de dépendance avec un autre îlot de l'entité fonctionnelle est supprimé de celle-ci,
- l'hypothèse que la relation de dépendance supprimée ne soit pas interne à l'entité fonctionnelle est équivalente au cas où un îlot n'ayant pas de relation de dépendance avec un autre îlot de l'entité fonctionnelle est supprimé de celle-ci.

Dans l'illustration choisie de la vue fonctionnelle avec les îlots représentés dans la **Figure 31** et les séquences de Se représentées dans la **Figure 32**, quatre entités fonctionnelles sont envisageables :

- $E = \left\{ \begin{array}{l} \text{Sécurité de la messagerie - Filtrage de message,} \\ \text{Sécurité de la messagerie - Gérer les règles de filtrage} \end{array} \right\}$
 $\Rightarrow N_1(E, Se) = 1$ pour l'îlot cible de l'interaction *Recevoir un message* de type réponse,
et $N_3(E, Se) = 1$ pour les trois instances d'îlots participantes à l'enchaînement des interactions *Créer les règles de filtrage de message* de type requête et *Créer les règles de filtrage de message* de type réponse.

$$\Rightarrow MPRI(E, Sc, Se) = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se)}{\sum_{i \geq 1} N_i(E, Se)} = \frac{(1*0) + (2*1)}{1+0+1} = \frac{2}{2} = 1$$
- $E = \{ \text{Sécurité de la messagerie - Filtrage de message, Réception d'un message} \}$
 $\Rightarrow N_1(E, Se) = 1$ pour l'îlot source de l'interaction *Créer les règles de filtrage* de type requête,
et $N_3(E, Se) = 1$ pour les trois instances d'îlots participantes à l'enchaînement des interactions *Recevoir un message* de type requête et *Recevoir un message* de type réponse.

$$\Rightarrow MPRI(E, Sc, Se) = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se)}{\sum_{i \geq 1} N_i(E, Se)} = \frac{(1*0) + (2*1)}{1+0+1} = \frac{2}{2} = 1$$
- $E = \{ \text{Sécurité de la messagerie - Gérer les règles de filtrage, Réception d'un message} \}$
 $\Rightarrow N_1(E, Se) = 2$ pour l'instance d'îlot ciblée par l'interaction *Créer les règles de filtrage de message* de type requête et l'instance d'îlot ciblée par l'interaction *Recevoir un message* de type requête

$$\Rightarrow MPRI(E, Sc, Se) = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se)}{\sum_{i \geq 1} N_i(E, Se)} = 0$$
- $E = \left\{ \begin{array}{l} \text{Sécurité de la messagerie - Filtrage de message,} \\ \text{Sécurité de la messagerie - Gérer les règles de filtrage, Réception d'un message} \end{array} \right\}$
 $N_5(E, Se) = 1$ pour les cinq instances d'îlots participantes à l'enchaînement des interactions *Créer les règles de filtrage de message* de type requête et *Créer les règles de filtrage de message* de type réponse, *Recevoir un message* de type requête et *Recevoir un message* de type réponse.

$$\Rightarrow MPRI(E, Sc, Se) = \frac{\sum_{i \geq 2} (i-1) * N_i(E, Se)}{\sum_{i \geq 1} N_i(E, Se)} = \frac{(1*0) + (2*0) + (3*0) + (4*1)}{0+0+0+0+1} = 4$$

Parmi toutes ces entités fonctionnelles, seule $E = \{ \text{Sécurité de la messagerie - Gérer les règles de filtrage, Réception d'un message} \}$ a une pertinence de regroupement de ses îlots fonctionnels nulle, puisqu'il n'existe pas de relation de dépendance entre ces îlots dans Se.

3.5.3 Seuil d'insertion d'un îlot fonctionnel dans une entité fonctionnelle

Le résultat de la conception des quartiers est alors celui de l'analyse de toutes les entités fonctionnelles possibles de pertinence non nulle. Afin de faciliter l'analyse, un seuil d'insertion d'un îlot fonctionnel dans une entité fonctionnelle est proposé [INPI – 05929] [INPI – 06471].

Si la pertinence résultante de l'insertion d'un îlot dans une entité fonctionnelle est supérieure au seuil d'insertion de l'îlot dans cette entité fonctionnelle, alors l'entité fonctionnelle résultante de l'insertion est un quartier fonctionnel potentiel. Sinon, l'entité fonctionnelle n'est pas un quartier fonctionnel potentiel.

Le seuil d'insertion d'un îlot fonctionnel dans une entité fonctionnelle est fondé sur un nombre minimum d'interactions entre un îlot fonctionnel de chaque séquence de i îlots fonctionnels de E et l'îlot fonctionnel inséré I . Ce nombre minimum d'interactions est calculé à partir du nombre de scénarios où apparaissent des séquences de i îlots fonctionnels de E et du nombre de scénarios où apparaît I .

L'évaluation du seuil d'insertion d'un îlot fonctionnel dans une entité fonctionnelle est un outil mis à la disposition de l'urbaniste fonctionnel afin d'optimiser la conception des quartiers fonctionnels d'une zone du PLU fonctionnel.

Définition : La mesure MSIE du seuil d'insertion d'un îlot fonctionnel I dans une entité fonctionnelle E à partir d'un ensemble Se de séquences d'instances d'îlot fonctionnel réalisées par un ensemble Sc de scénarios est telle que :

$$MSIE(E, I, Sc, Se) = \left(\frac{N_1(\{I\}, Sc)}{N_1(\{I\}, Se)} * \frac{\left(\sum_{i \geq 1} i * (N_i(E, Sc)) \right)}{\left(\sum_{i \geq 1} N_i(E, Se) \right)} \right) + \left(\frac{\sum_{i \geq 2} (i - 1) * (N_i(E, Se) - N_i(E, Sc))}{N_1(\{I\}, Se) * \sum_{i \geq 1} N_i(E, Se)} \right)$$

Équation 3-5

avec :

- Se , l'ensemble des séquences d'îlots instanciés lors du déroulement de scénarios de cas d'utilisation du système,
- Sc , l'ensemble des scénarios de cas d'utilisation du système,
- $N_i(E, Se)$, le nombre de séquences de i instances d'îlots fonctionnels appartenant à E extraites de Se ,
- $N_i(E, Sc)$, le nombre de scénarios distincts comportant une séquence de i instances d'îlots fonctionnels appartenant à E extraites Sc .

De la même façon que pour la pertinence du regroupement d'îlots fonctionnels dans une même entité fonctionnelle, les séquences d'instances d'îlots fonctionnels prises en compte dans le calcul de MSIE sont les séquences internes à l'entité fonctionnelle de longueur maximale.

Dans l'illustration choisie de la vue fonctionnelle avec les îlots représentés dans la **Figure 31** et les séquences de Se du seul scénario de Sc représenté dans la **Figure 32**, quatre insertions d'îlot fonctionnel dans une entité fonctionnelle sont envisageables (seule est écartée l'entité fonctionnelle résultante de l'insertion dont la pertinence est nulle dans l'illustration du §3.5.2) :

- Soient $E = \{\text{Sécurité de la messagerie - Filtrage de message}\}$ et
 $I = \text{Sécurité de la messagerie - Gérer les règles de filtrage}$
 $\Rightarrow N_1(\{I\}, Sc) = 1$ et $N_1(\{I\}, Se) = 1$ pour les instances de l'îlot I, et,
 $N_1(E, Sc) = 1$ et $N_1(E, Se) = 3$ pour les instances de l'îlot *Sécurité de la messagerie - Filtrage de message*.
 $\Rightarrow MSIE(E, I, Se, Sc) =$

$$\left(\frac{N_1(\{I\}, Sc)}{N_1(\{I\}, Se)} * \frac{\left(\sum_{i \geq 1} i * (N_i(E, Sc)) \right)}{\left(\sum_{i \geq 1} N_i(E, Se) \right)} \right) + \left(\frac{\sum_{i \geq 2} (i-1) * (N_i(E, Se) - N_i(E, Sc))}{N_1(\{I\}, Se) * \sum_{i \geq 1} N_i(E, Se)} \right) =$$

$$\frac{N_1(\{I\}, Sc)}{N_1(\{I\}, Se)} * \frac{N_1(E, Sc)}{N_1(E, Se)} = \frac{1}{1} * \frac{1}{3} = \frac{1}{3}$$

Remarque : Le seuil est identique pour
 $E = \{\text{Sécurité de la messagerie - Gérer les règles de filtrage}\}$ et
 $I = \text{Sécurité de la messagerie - Filtrage de message}$.

La pertinence de $E \cup \{I\}$ évaluée dans le §3.5.2 est
 $MPRI(E \cup \{I\}, Sc, Se) = 1 > \frac{1}{3} = MSIE(E, I, Se, Sc) \Rightarrow$ l'entité fonctionnelle
 $E \cup \{I\}$ est mémorisée comme un quartier fonctionnel potentiel.

- Soient $E = \{\text{Sécurité de la messagerie - Filtrage de message}\}$ et
 $I = \text{Réception d'un message}$
 $\Rightarrow N_1(\{I\}, Sc) = 1$ et $N_1(\{I\}, Se) = 1$ pour les instances de l'îlot I, et,
 $N_1(E, Sc) = 1$ et $N_1(E, Se) = 3$ pour les instances de l'îlot *Sécurité de la messagerie - Filtrage de message*.
 $\Rightarrow MSIE(E, I, Se, Sc) =$

$$\left(\frac{N_1(\{I\}, Sc)}{N_1(\{I\}, Se)} * \frac{\left(\sum_{i \geq 1} i * (N_i(E, Sc)) \right)}{\left(\sum_{i \geq 1} N_i(E, Se) \right)} \right) + \left(\frac{\sum_{i \geq 2} (i-1) * (N_i(E, Se) - N_i(E, Sc))}{N_1(\{I\}, Se) * \sum_{i \geq 1} N_i(E, Se)} \right) =$$

$$\frac{N_1(\{I\}, Sc)}{N_1(\{I\}, Se)} * \frac{N_1(E, Sc)}{N_1(E, Se)} = \frac{1}{1} * \frac{1}{3} = \frac{1}{3}$$

Remarque : Le seuil est identique pour $E = \{\text{Réception d'un message}\}$ et
 $I = \text{Sécurité de la messagerie - Filtrage de message}$.

La pertinence de $E \cup \{I\}$ évaluée dans le §3.5.2 est
 $MPRI(E \cup \{I\}, Sc, Se) = 1 > \frac{1}{3} = MSIE(E, I, Se, Sc) \Rightarrow$ l'entité fonctionnelle
 $E \cup \{I\}$ est mémorisée comme un quartier fonctionnel potentiel.

- Soient $E = \left\{ \begin{array}{l} \text{Sécurité de la messagerie - Filtrage de message,} \\ \text{Sécurité de la messagerie - Gérer les règles de filtrage} \end{array} \right\}$ et

I = Réception d'un message

$\Rightarrow N_1(\{I\}, Sc) = 1$ et $N_1(\{I\}, Se) = 1$ pour les instances de l'îlot I, et,

$N_1(E, Sc) = 1$ et $N_1(E, Se) = 1$ ainsi que $N_3(E, Sc) = 1$ et $N_3(E, Se) = 1$ pour les instances des îlots fonctionnels *Sécurité de la messagerie - Filtrage de message* et *Sécurité de la messagerie - Gérer les règles de filtrage*.

$\Rightarrow MSIE(E, I, Se, Sc) =$

$$\left(\frac{N_1(\{I\}, Sc)}{N_1(\{I\}, Se)} * \frac{\left(\sum_{i \geq 1} i * (N_i(E, Sc)) \right)}{\left(\sum_{i \geq 1} N_i(E, Se) \right)} \right) + \left(\frac{\sum_{i \geq 2} (i-1) * (N_i(E, Se) - N_i(E, Sc))}{N_1(\{I\}, Se) * \sum_{i \geq 1} N_i(E, Se)} \right) =$$

$$\left(\frac{1}{1} * \frac{(1 * 1) + (2 * 0) + (3 * 1)}{1 + 0 + 1} \right) + \left(\frac{2 * (1 - 1)}{1 * (1 + 0 + 1)} \right) = 2$$

La pertinence de $E \cup \{I\}$ évaluée dans le §3.5.2 est $MPRI(E \cup \{I\}, Sc, Se) = 4 > 2 = MSIE(E, I, Se, Sc) \Rightarrow$ l'entité fonctionnelle $E \cup \{I\}$ est mémorisée comme un quartier fonctionnel potentiel.

- Soient $E = \{\text{Sécurité de la messagerie - Filtrage de message, Réception d'un message}\}$ et I = Sécurité de la messagerie - Gérer les règles de filtrage

$\Rightarrow N_1(\{I\}, Sc) = 1$ et $N_1(\{I\}, Se) = 1$ pour les instances de l'îlot I, et,

$N_1(E, Sc) = 1$ et $N_1(E, Se) = 1$ ainsi que $N_3(E, Sc) = 1$ et $N_3(E, Se) = 1$ pour les instances des îlots fonctionnels *Sécurité de la messagerie - Filtrage de message* et *Réception d'un message*.

$\Rightarrow MSIE(E, I, Se, Sc) =$

$$\left(\frac{N_1(\{I\}, Sc)}{N_1(\{I\}, Se)} * \frac{\left(\sum_{i \geq 1} i * (N_i(E, Sc)) \right)}{\left(\sum_{i \geq 1} N_i(E, Se) \right)} \right) + \left(\frac{\sum_{i \geq 2} (i-1) * (N_i(E, Se) - N_i(E, Sc))}{N_1(\{I\}, Se) * \sum_{i \geq 1} N_i(E, Se)} \right) =$$

$$\left(\frac{1}{1} * \frac{(1 * 1) + (2 * 0) + (3 * 1)}{1 + 0 + 1} \right) + \left(\frac{2 * (1 - 1)}{1 * (1 + 0 + 1)} \right) = 2$$

La pertinence de $E \cup \{I\}$ évaluée dans le §3.5.2 est $MPRI(E \cup \{I\}, Sc, Se) = 4 > 2 = MSIE(E, I, Se, Sc) \Rightarrow$ l'entité fonctionnelle $E \cup \{I\}$ est mémorisée comme un quartier fonctionnel potentiel.

3.5.4 Algorithme bactériologique de conception des quartiers fonctionnels à partir des entités fonctionnelles

Un algorithme de conception des quartiers fonctionnels existe déjà à partir de la pertinence d'une entité fonctionnelle et du seuil de regroupement d'entités fonctionnelles [INPI – 06471]. Le principe est de représenter le regroupement d'entités fonctionnelles dans une structure arborescente. Chaque nœud de l'arbre est une entité fonctionnelle. Un arbre binaire d'entités fonctionnelles est alors conçu à partir des meilleurs regroupements d'entités fonctionnelles. Les regroupements sont classés à partir du rapport entre la pertinence de l'entité fonctionnelle résultante du regroupement et le seuil de regroupement des deux entités fonctionnelles cibles du regroupement. La difficulté lors de cette conception arborescente est d'éviter les minima locaux. En effet, lors d'une égalité des gains, un choix entre les regroupements d'entités fonctionnelles ayant le même gain est réalisé de façon aléatoire. Ce choix peut faire disparaître dans la suite de l'algorithme des entités fonctionnelles pertinentes.

Afin d'éviter les minima locaux, une solution est d'utiliser un algorithme génétique [Goldberg, 1989]. En informatique, l'algorithmique génétique permet, par exemple, de regrouper des composants d'un système afin d'optimiser leur configuration [Zhang, 2000]. Dans le cas de regroupement d'îlots, les entités fonctionnelles forment en effet une population qui évolue du fait d'insertions successives d'îlots dans une entité ou de suppression d'îlots dans une entité. L'insertion d'un îlot fonctionnel dans une entité fonctionnelle est synonyme d'une mutation d'un élément de la population. Au cours de ces mutations successives, une sélection des meilleures entités fonctionnelles permet de concevoir les quartiers fonctionnels.

Le besoin de l'urbaniste fonctionnel est d'obtenir une liste de quartiers fonctionnels potentiels afin de choisir parmi ceux-ci les plus pertinents quant au PLU fonctionnel cible. Son choix d'expert fonctionnel du SI peut être différent de celui de l'ensemble de quartiers fonctionnels disjoints recouvrant sa zone et possédant la pertinence la plus élevée (cf. §3.5.2). Une solution est de mémoriser les meilleurs regroupements et de supprimer les plus faibles. C'est la raison pour laquelle, la conception des quartiers fonctionnels est supportée par une approche bactériologique. Cette approche fondée sur le comportement des bactéries a été mise en œuvre pour l'automatisation des tests de logiciel [Baudry1, 2005]. Le principe retenu est de définir une bactérie par une entité fonctionnelle, soit un regroupement d'îlots fonctionnels.

La qualité d'une bactérie est corrélée au couplage des îlots fonctionnels qui la constituent lors du déroulement de scénarios de cas d'utilisation de la zone fonctionnelle :

- une bactérie est d'autant meilleure que le couplage entre ses îlots fonctionnels est fort,
- une mauvaise bactérie est une bactérie où les îlots fonctionnels ont un couplage nul.

Les critères de qualité des bactéries sont définis par des fonctions bactériologiques [Baudry2, 2005] adaptées au regroupement d'îlots fonctionnels.

3.5.4.1 Fonctions bactériologiques

Si Λ est l'ensemble des entités fonctionnelles regroupant un ou plusieurs îlots fonctionnels, la fonction d'utilité F d'un ensemble d'entités fonctionnelles E_i , telles que $\forall i, E_i \in \Lambda$, permet d'évaluer la qualité de cet ensemble. F est définie à partir de la pertinence MPRI d'une entité fonctionnelle (cf. §3.5.2). Cette pertinence est fonction de l'ensemble Se des séquences d'îlots instanciées par un des scénarios de cas d'utilisation de la zone fonctionnelle et de l'ensemble Sc des scénarios de cas d'utilisation du système.

$$F : 2^\Lambda \rightarrow \mathbb{R}^+$$

$$F\left(\bigcup_i E_i\right) = \sum_i MPRI(E_i, Sc, Se)$$

La fonction d'utilité relative f_ρ mesurant la qualité d'une entité fonctionnelle E par rapport à l'ensemble Λ est définie par rapport à F :

$$f_\rho : \Lambda \times 2^\Lambda \rightarrow \mathfrak{R}^+$$

$$f_\rho \left(E, \left(\bigcup_i E_i \right) \right) = F \left(E \bigcup \left(\bigcup_i E_i \right) \right) - F \left(\bigcup_i E_i \right)$$

La définition de f_ρ à valeur dans \mathfrak{R}^+ est due à la croissance de F . En effet,

$$\forall E_i \in \Lambda, \forall E \in \Lambda \Rightarrow F \left(E \bigcup \left(\bigcup_i E_i \right) \right) - F \left(\bigcup_i E_i \right) = MPRI(E, Sc, Se) \geq 0$$

La fonction de mémorisation f_v permet d'associer à une entité fonctionnelle E résultante de l'insertion d'un îlot fonctionnel I dans une entité fonctionnelle E_0 le fait qu'elle soit à mémoriser ou non en tant que quartier fonctionnel. La fonction f_v est définie par rapport au seuil MSIE (cf. §3.5.3) du regroupement dont est issue l'entité fonctionnelle:

$$f_v : \Lambda \rightarrow (0,1)$$

$$f_v(E) = 1 \Leftrightarrow MPRI(E_0 \bigcup \{I\}, Sc, Se) \geq MSIE(E_0, I, Sc, Se)$$

$$f_v(E) = 0 \Leftrightarrow MPRI(E_0 \bigcup \{I\}, Sc, Se) < MSIE(E_0, I, Sc, Se)$$

La fonction de filtrage f_ϕ supprime les entités fonctionnelles dont l'utilité relative est nulle. Ceci est équivalent à supprimer les entités fonctionnelles dont la pertinence est nulle, c'est-à-dire celles dont les îlots fonctionnels n'ont pas de relations de dépendance instanciées durant les scénarios de Sc .

La fonction de mutation f_μ permet de modifier une entité fonctionnelle en lui insérant un seul îlot fonctionnel. La fonction de mutation ainsi définie respecte le fait de ne modifier que légèrement une entité fonctionnelle.

3.5.4.2 Algorithme bactériologique

A l'aide des fonctions bactériologiques définies dans le §3.5.4.1, l'algorithme bactériologique est structuré dans un cycle incrémental constitué des quatre activités suivantes :

- la mutation où un îlot est, soit inséré dans une entité fonctionnelle, soit supprimé d'une entité fonctionnelle,
- le classement des entités fonctionnelles,
- la mémorisation des meilleures entités fonctionnelles,
- le filtrage permettant de supprimer les mauvaises entités fonctionnelles.

La procédure est décrite dans le diagramme de tâches de la **Figure 37**.

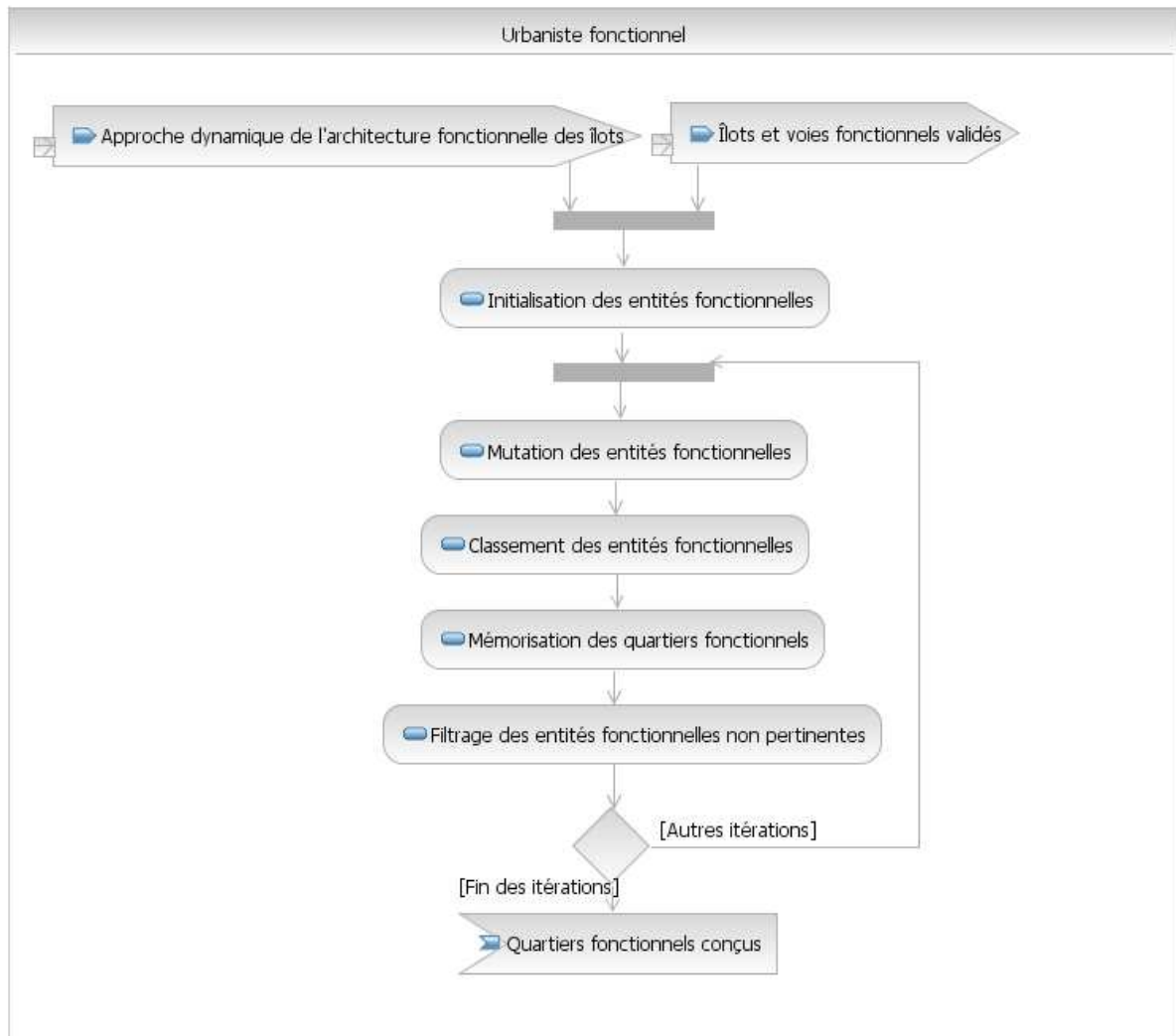


Figure 37 : Procédure de conception automatisée des quartiers fonctionnels

Chacune des activités de la procédure de conception automatisée des quartiers fonctionnels est détaillée dans l'algorithme suivant où

- NE_k représente le nombre d'entités fonctionnelles résultantes de la k° itération,
- I_i est un îlot fonctionnel,
- $E_{k,j}$ est une entité fonctionnelle non vide résultante de la k° itération.

```

/* Initialisation des entités fonctionnelles à partir des  $NE_0$  îlots fonctionnels  $I_i$  */
Pour i de 1 à  $NE_0$  {
     $E_{0,i} = \{I_i\}$ ;
}

```

```

/* k° itération */
/* Mutation */
Pour j de 1 à NEk-1 {
  Si  $i \in [1, NE_0]$  tiré au sort et si le taux de mutation n'est pas dépassé

  Alors  $E_{k,j} = f_{\mu} \left( E_{k-1,j}, \bigcup_i E_{0,i} \right) = E_{k-1,j} \bigcup E_{0,i}$  ,

  Si  $E_{0,i} \in E_{k-1,i}$ 

  Alors  $E_{k,i} = f_{\mu} \left( E_{k-1,i}, \bigcup_i E_{0,i} \right) = E_{k-1,i} - E_{0,i}$ 

  Finsi
Finsi
}
/* Remarques:  $0 \leq Card(E_{k,j}) \leq Card(E_{k-1,j}) + 1$  et  $\forall j, \forall l, E_{k,j} \cap E_{k,l} = \{\emptyset\}$  */
/* Classement */
Pour j de 1 à NEk {
  Pour l de (j + 1) à NEk {
    Si  $f_{\rho} \left( E_{k,j}, \bigcup_{i \leq k,m} E_{i,m} \right) \leq f_{\rho} \left( E_{k,l}, \bigcup_{i \leq k,m} E_{i,m} \right)$ 

    Alors  $E_{k,j} \leq E_{k,l}$ 

    Sinon  $E_{k,j} > E_{k,l}$ 

    Finsi
  }
}
/* Mémorisation */
Pour j de 1 à NEk {
  Si  $E_{k,j} = f_{\mu} \left( E_{k-1,j}, \bigcup_i E_{0,i} \right) = E_{k-1,j} \bigcup E_{0,i}$  ,

  Alors
  Si  $f_v(E_{k-1,j} \bigcup E_{0,i}) = 1$ 
  Alors  $E_{k,j}$  est mémorisée
  Sinon  $E_{k,j}$  n'est pas mémorisée
  Finsi
Finsi
}
/* Filtrage */
Pour j de 1 à NEk {
  Si  $f_{\rho} \left( E_{k,j}, \bigcup_{i \leq k,m} E_{i,m} \right) \neq 0$ 

  Alors  $E_{k,j}$  n'est pas supprimée
  Sinon  $E_{k,j}$  est supprimée
  Finsi
}

```

L'algorithme bactériologique permettant de concevoir les quartiers fonctionnels est codé avec le langage SmartQVT [SmartQVT, web] qui implémente le langage QVT-Operational, partie de la norme QVT (Query / View / Transformation) [QVT, web] définie par l'OMG (Open Management Group) comme langage de l'approche MDE.

3.6 Expérimentation avec le SI supportant le processus métier d'usage d'un service télécom

3.6.1 Démarche appliquée à la zone Messagerie

Dans le PLU fonctionnel conçu par un groupe d'experts du SI supportant le processus de l'usage d'un service télécom à France Télécom, il existe actuellement une dizaine de zones fonctionnelles, une cinquantaine d'îlots fonctionnels et une centaine de voies entre îlots. L'illustration choisie est la zone fonctionnelle *Messagerie* dont l'urbaniste est actuellement Francis Alizon de France Télécom R&D.

3.6.1.1 Conception experte des zones, des îlots des parcelles et des voies fonctionnels

Les îlots fonctionnels de la zone fonctionnelle *Messagerie* sont représentés dans l'extrait de ce PLU fonctionnel **Figure 38**. De l'extrait de la zone *Messagerie*, seules les parcelles prises en compte dans l'approche dynamique sont étudiées.

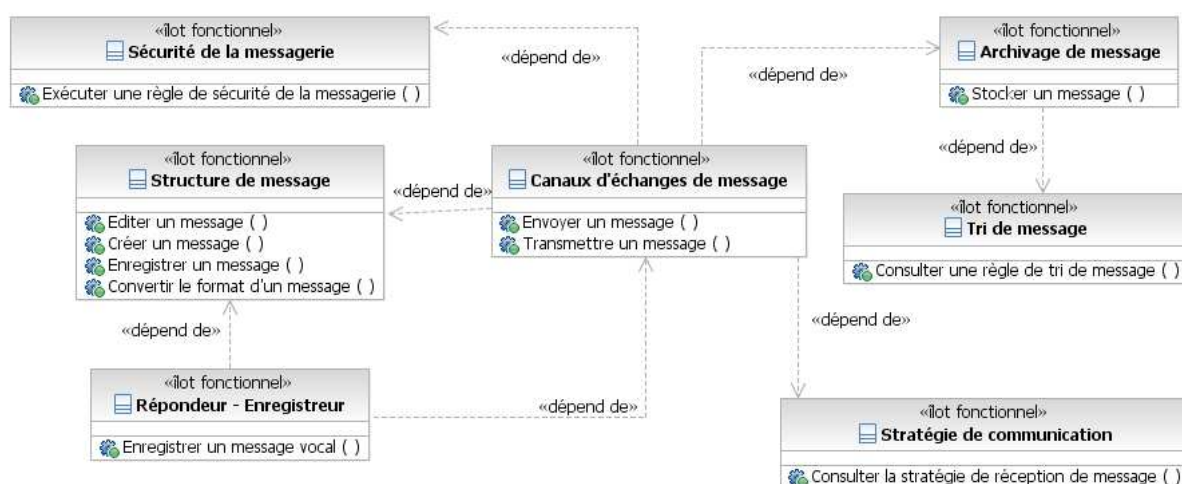


Figure 38 : Extrait des îlots, des parcelles et des voies fonctionnels de la zone Messagerie conçus par l'expert

Les diagrammes de séquence UML significatifs de l'approche dynamique pour la zone *Messagerie* sont référencés pour chaque scénario dans le Tableau 6.

Scénario de la zone <i>Messagerie</i>	Figure
Envoyer un email	Figure 39
Envoyer un message vocal converti en MMS	Figure 40
Envoyer un message vocal	Figure 41
Recevoir un email infecté	Figure 42
Notifier par email une alerte de boîte de messagerie	Figure 43
Notifier par email la lecture d'un email reçu	Figure 44

Tableau 6 : Références des scénarios illustrant la zone Messagerie.

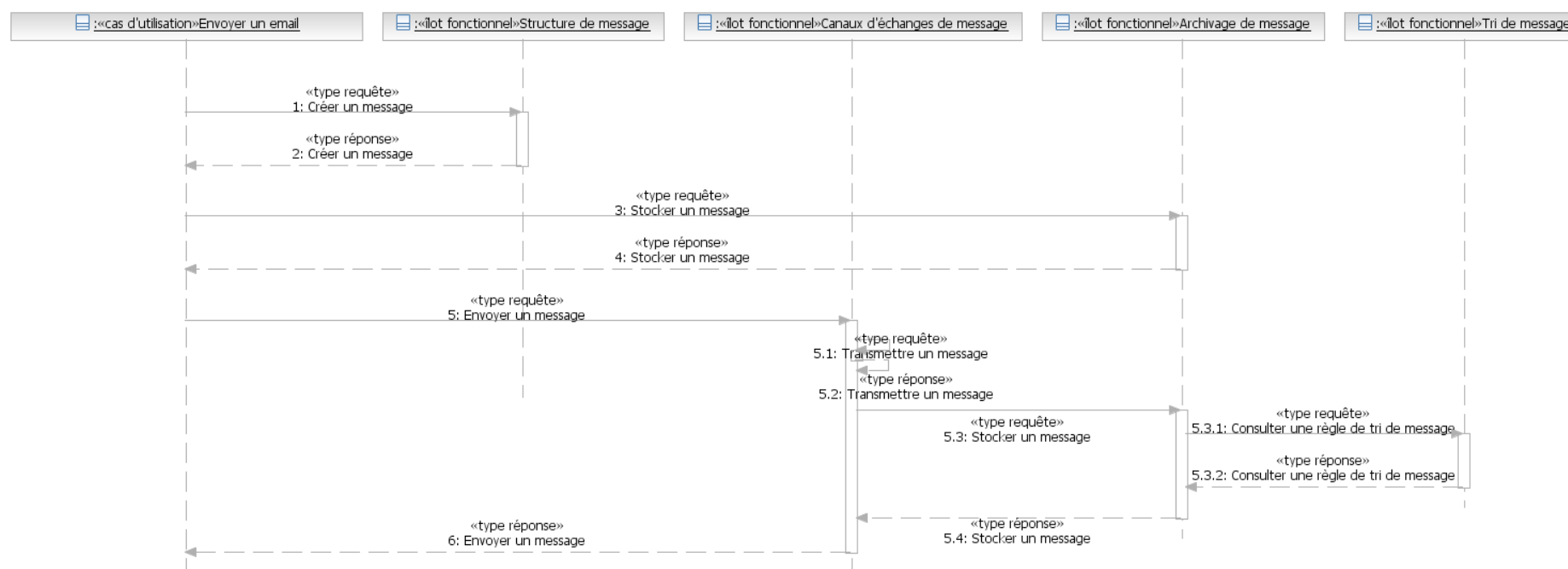


Figure 39 : Scénario d'envoi de message

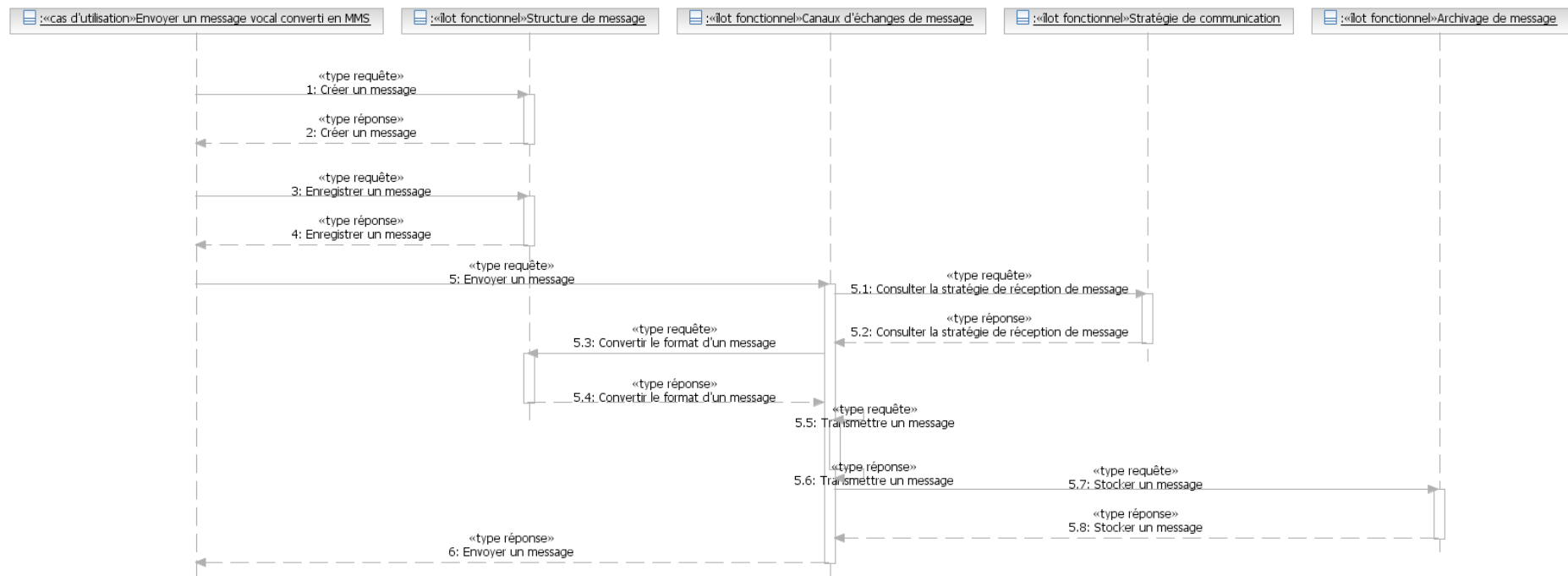


Figure 40 : Scénario d'envoi d'un message vocal converti en MMS

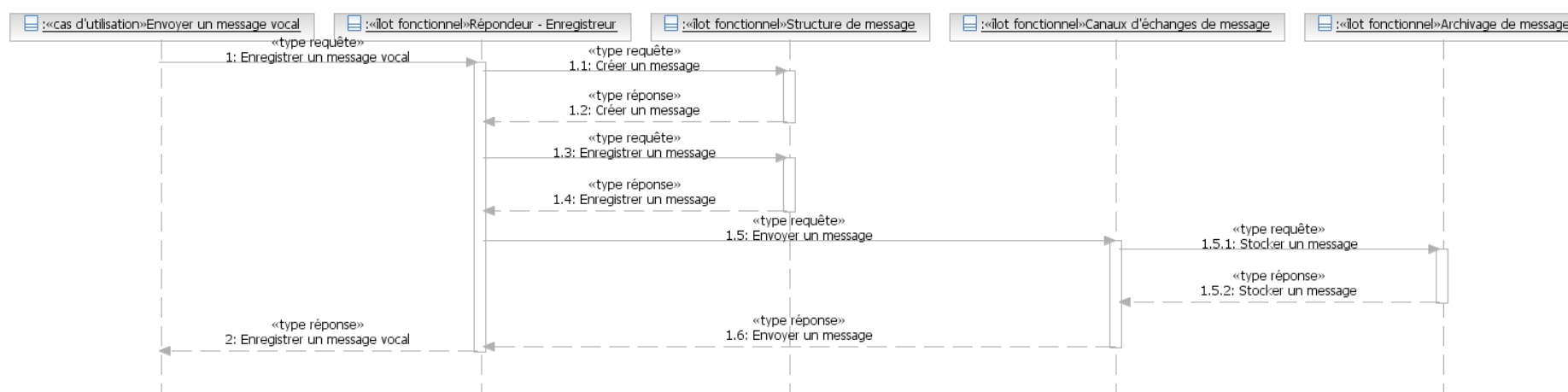


Figure 41 : Scénario d'envoi d'un message vocal

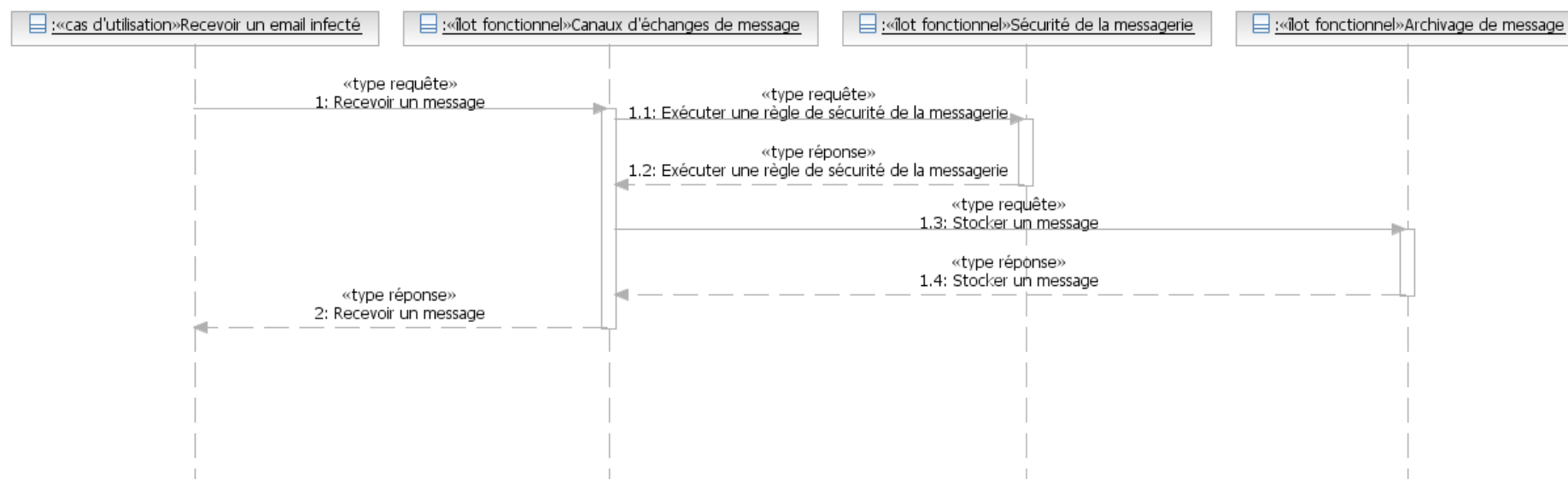


Figure 42 : Scénario de réception d'un email infecté

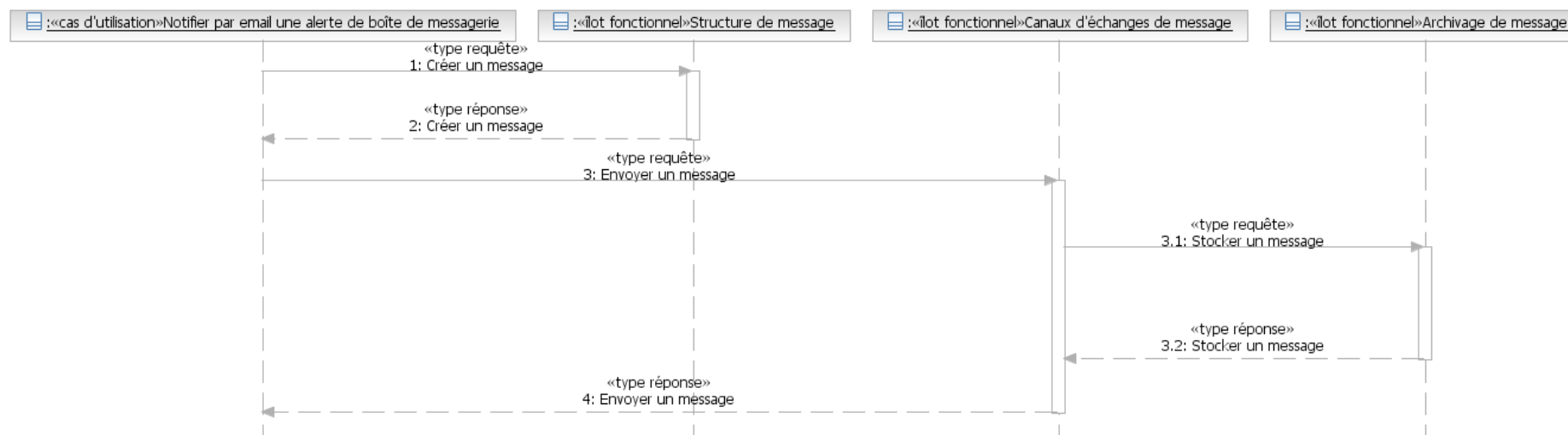


Figure 43 : Scénario de notification par email d'une alerte de boîte de messagerie

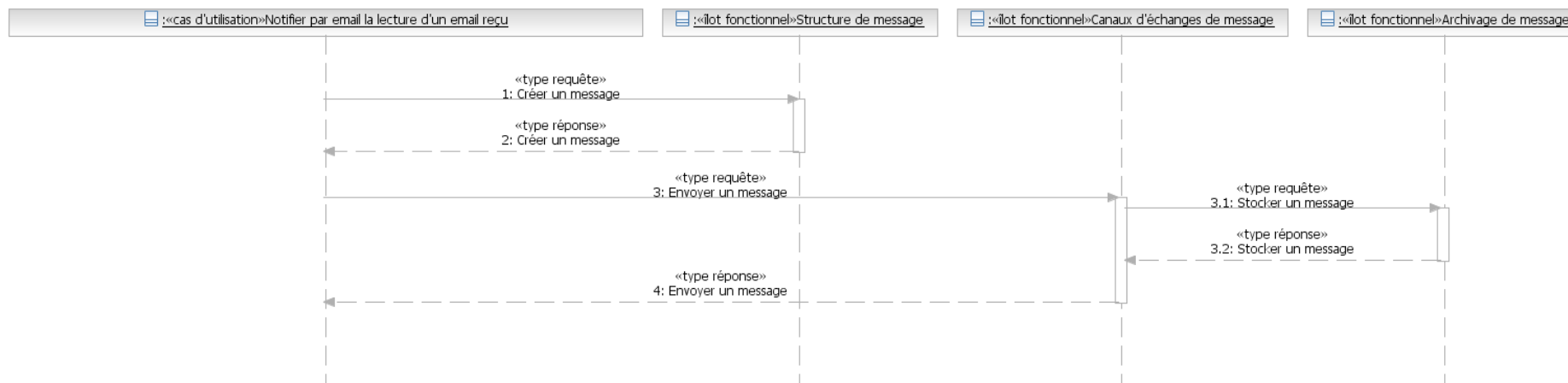


Figure 44 : Scénario de notification par email de la lecture d'un email reçu

3.6.1.2 Typage des îlots fonctionnels

Le typage des îlots fonctionnels est fourni par l'urbaniste responsable de la zone *Messagerie*. D'après l'Équation 3-1, la formulation de ce typage est la suivante :

$$\begin{aligned} t(\text{Sécurité de la messagerie}) &= \{Flux\} \\ t(\text{Canaux d'échanges de message}) &= \{Flux\} \\ t(\text{Structure de message}) &= \{Stock\} \\ t(\text{Répondeur - Enregistreur}) &= \{Flux, Stock\} \\ t(\text{Archivage de message}) &= \{Stock\} \\ t(\text{Tri de message}) &= \{Stock\} \\ t(\text{Stratégie de communication}) &= \{Stock\} \end{aligned}$$

Seul, l'îlot fonctionnel *Répondeur - Enregistreur* est multi-typé. En effet la parcelle fonctionnelle *Enregistrer un message vocal* recouvre :

- une parcelle fonctionnelle *Envoyer un message vocal* qui définit un îlot de type flux et,
 - une parcelle fonctionnelle *Créer un message vocal* qui définit un îlot de type stock.
- Ceci est équivalent, d'après l'Équation 3-2 à :

$$s(\text{Répondeur - Enregistreur}) = \left\{ \begin{array}{l} (\text{Répondeur - Enregistreur - Envoi de message vocal}, Flux), \\ (\text{Répondeur - Enregistreur - Création de message vocal}, Stock) \end{array} \right\}$$

De plus, une relation de dépendance est créée de l'îlot fonctionnel *Répondeur - Enregistreur - Envoi de message vocal* vers l'îlot fonctionnel *Répondeur - Enregistreur - Création de message vocal* conformément à la Règle d'urbanisme - Flux et stock.

3.6.1.3 Conception automatisée des îlots et voies fonctionnels

La dépendance de l'îlot fonctionnel *Répondeur - Enregistreur* vis-à-vis de l'îlot fonctionnel *Canaux d'échanges de message* est transformée conformément à la Règle d'urbanisme - Scission îlot multi-typé en :

- une dépendance de l'îlot fonctionnel *Répondeur - Enregistreur - Envoi de message vocal* de type flux vers l'îlot fonctionnel *Canaux d'échanges de message* de type flux,
- une dépendance de l'îlot fonctionnel *Répondeur - Enregistreur - Envoi de message vocal* de type flux vers l'îlot fonctionnel *Structure de message* de type stock,
- une dépendance de l'îlot fonctionnel *Répondeur - Enregistreur - Création de message vocal* de type stock vers l'îlot fonctionnel *Structure de message* de type stock.

Toujours conformément à cette règle, une dépendance de l'îlot fonctionnel *Répondeur - Enregistreur - Création de message vocal* de type stock vers l'îlot fonctionnel *Canaux d'échanges de message* de type flux devrait être ajoutée au modèle d'îlots fonctionnels. Cet ajout est impossible du fait de l'absence du respect de la Règle d'urbanisme - Flux et stock.

Le résultat de la transformation du modèle des îlots fonctionnels de la zone *Messagerie* conformément à la Règle d'urbanisme - Scission îlot multi-typé et à la Règle d'urbanisme - Flux et stock est représentée **Figure 45**.

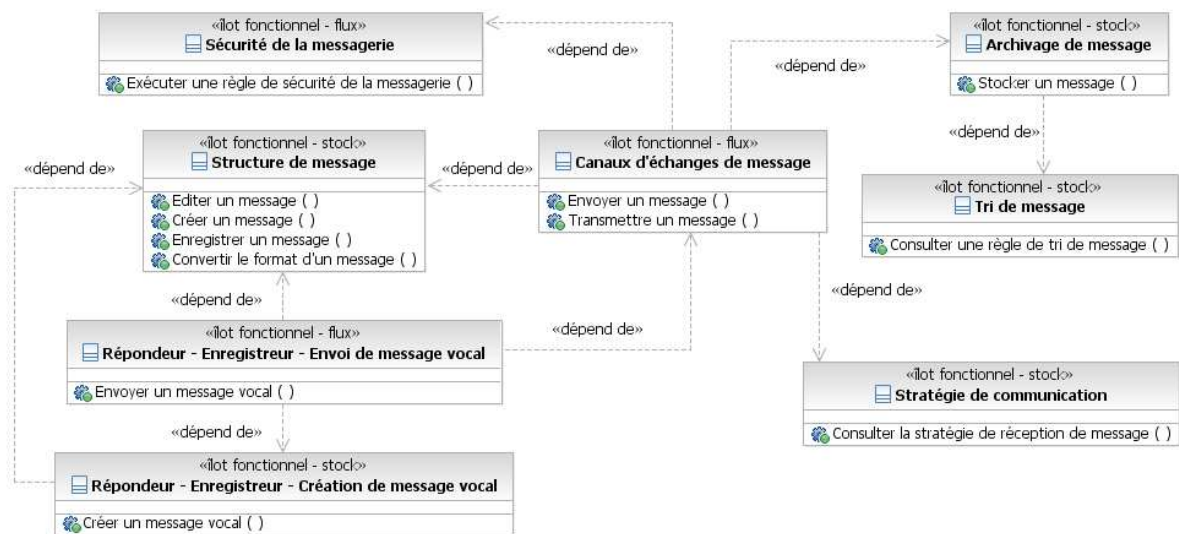


Figure 45 : Extrait des îlots et des voies fonctionnels de la zone messagerie contraints par le typage des îlots

Seul le scénario d'envoi d'un message vocal est modifié suite à l'évolution du modèle d'îlots fonctionnels. La parcelle *Enregistrer un message vocal* est scindée en deux interactions conformes au modèle d'îlots fonctionnels de la **Figure 45** :

- *Envoyer un message vocal* du scénario d'envoi de message vocal vers l'îlot fonctionnel *Répondeur – Enregistreur – Envoi de message vocal*
- *Créer un message vocal* de l'îlot fonctionnel *Répondeur – Enregistreur – Envoi de message vocal* vers l'îlot fonctionnel *Répondeur – Enregistreur – Création de message vocal*

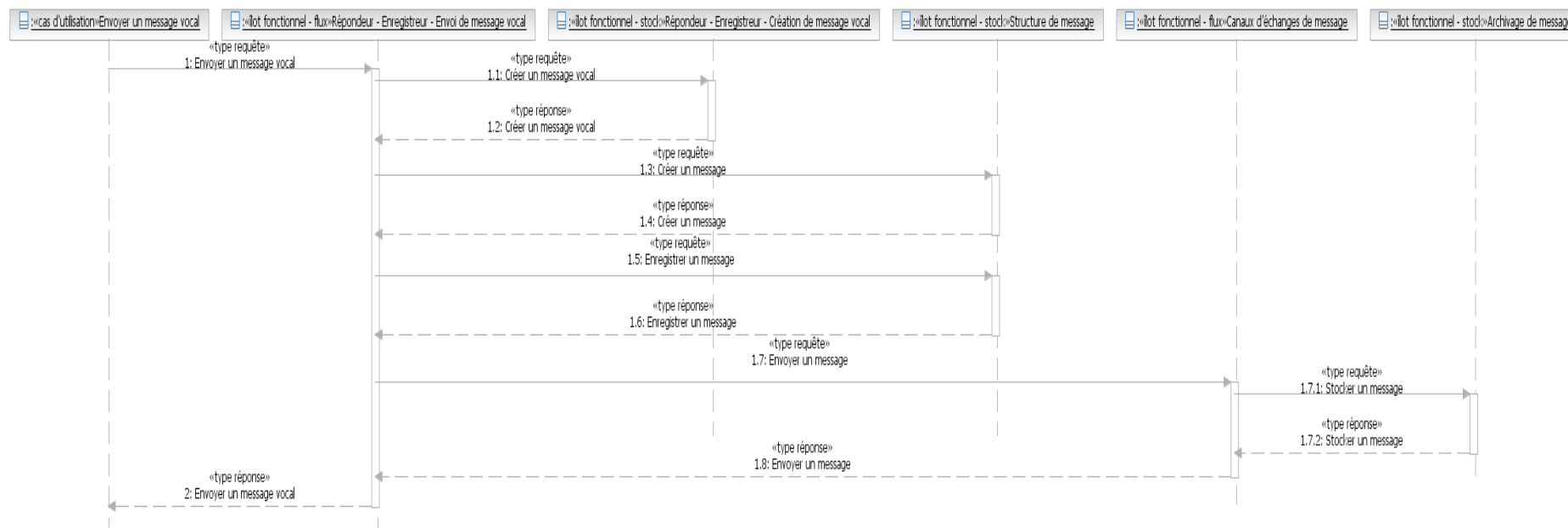


Figure 46 : Scénario d'envoi d'un message vocal contraint par le typage des îlots

3.6.1.4 Mesure de l'alignement avec la vue métier

Le processus métier d'envoi de message électronique est décrit dans le diagramme d'activités UML de la **Figure 19**.

L'alignement des parcelles fonctionnelles avec la vue métier proposé par l'urbaniste fonctionnel responsable de la zone *Messagerie* est reporté dans le Tableau 7.

Parcelle fonctionnelle	Activité métier
Exécuter une règle de sécurité de la messagerie	<i>Envoyer un message</i>
Envoyer un message	<i>Envoyer un message</i>
	<i>Délivrer un message</i>
Recevoir un message	<i>Délivrer un message</i>
Transmettre un message	<i>Envoyer un message</i>
Editer un message	<i>Composer un message</i>
Créer un message	
Enregistrer un message	
Convertir le format d'un message	<i>Envoyer un message</i>
Créer un message vocal	<i>Composer un message</i>
Envoyer un message vocal	<i>Envoyer un message</i>
	<i>Délivrer un message</i>
Modifier un message vocal	<i>Composer un message</i>
Stocker un message	<i>Délivrer un message</i>
Consulter une règle de tri de message	
Consulter la stratégie de réception de message	<i>Envoyer un message</i>

Tableau 7 : Alignement des parcelles fonctionnelles de la zone Messagerie avec les activités métier du processus d'envoi de message électronique.

L'alignement des relations de dépendance entre parcelles fonctionnelles avec la vue métier est évalué dans le Tableau 8 de façon automatique. L'évaluation est faite pour chaque scénario.

Scénario	Relation de dépendance entre parcelles fonctionnelles	Relation de succession entre activités métier
Envoyer un email	Envoyer un message vers Transmettre un message	<i>Envoyer un message</i> vers elle-même
	Transmettre un message vers Stocker un message	Non alignée (relation de dépendance inverse)
	Stocker un message vers Consulter une règle de tri de message	<i>Délivrer un message</i> vers elle-même
Envoyer un message vocal converti en MMS	Envoyer un message vers Consulter la stratégie de réception de message	<i>Envoyer un message</i> vers elle-même
	Envoyer un message vers Convertir le format d'un message	<i>Envoyer un message</i> vers elle-même
	Envoyer un message vers Transmettre un message	<i>Envoyer un message</i> vers elle-même
	Transmettre un message vers Stocker un message	Non alignée (relation de dépendance inverse)

Envoyer un message vocal	Envoyer un message vocal vers Créer un message vocal	<i>Composer un message vers Envoyer un message</i>
	Envoyer un message vocal vers Créer un message	<i>Composer un message vers Envoyer un message</i>
	Envoyer un message vocal vers Enregistrer un message	<i>Composer un message vers Envoyer un message</i>
	Envoyer un message vocal vers Envoyer un message	<i>Envoyer un message vers elle-même</i>
	Envoyer un message vers Stocker un message	<i>Délivrer un message vers elle-même</i>
Recevoir un email infecté	Recevoir un message vers Exécuter une règle de sécurité de la messagerie	<i>Envoyer un message vers Délivrer un message</i>
	Recevoir un message vers Stocker un message	<i>Délivrer un message vers elle-même</i>
Notifier par email une alerte de boîte de messagerie	Envoyer un message vers Stocker un message	<i>Délivrer un message vers elle-même</i>
Notifier par email la lecture d'un email reçu	Envoyer un message vers Stocker un message	<i>Délivrer un message vers elle-même</i>

Tableau 8 : Alignement des relations de dépendance entre parcelles fonctionnelles de la zone Messagerie avec les relations de succession entre activités métier du processus d'envoi de message électronique.

La mesure MAFM de l'alignement de la vue fonctionnelle de la zone *Messagerie* par rapport à la vue métier est d'après l'Équation 3-3 :

$$MAFM(\text{zone Messagerie}) = \left(\frac{13-0}{13} \right) * \left(\frac{16-2}{16} \right) = \frac{7}{8}$$

puisque

- $N_p(\text{zone Messagerie}) = 13$ et $N_{nap}(\text{zone Messagerie}) = 0$ puisque les 13 parcelles sont alignées d'après le Tableau 7,
- $N_r(\text{zone Messagerie}) = 16$ et $N_{nar}(\text{zone Messagerie}) = 2$ puisque 2 relations de dépendance entre parcelles ne sont pas alignées d'après le Tableau 8.

Un bilan de cet alignement est envoyé à l'urbaniste fonctionnel responsable de la zone *Messagerie*. L'objectif est de lui permettre de valider la vue fonctionnelle de la zone ou de la faire évoluer.

Le bilan est le suivant:

- L'alignement de la vue fonctionnelle de la zone *Messagerie* avec le processus métier d'envoi de message électronique serait parfait (c'est-à-dire mesuré à 1) si la relation de dépendance entre
 - o la parcelle fonctionnelle *Transmettre un message* de l'îlot *Canaux d'échanges de message* et
 - o la parcelle fonctionnelle *Stocker un message* de l'îlot fonctionnel *Archivage de message*était inversée.

Cette inversion nécessiterait de scinder l'îlot fonctionnel *Canaux d'échange de message* en deux îlots :

- *Canaux d'échange de message – Envoi de message* défini par la parcelle fonctionnelle *Envoyer un message*
- *Canaux d'échange de message – Transmission de message* défini par la parcelle fonctionnelle *Transmettre un message*.

Dans ce cas, les trois relations de dépendance suivantes permettraient un alignement parfait :

- une relation de dépendance de l'îlot fonctionnel *Canaux d'échange de message – Envoi de message* vers l'îlot fonctionnel *Canaux d'échange de message – Transmission de message*,
- une relation de dépendance de l'îlot fonctionnel *Archivage de message* vers l'îlot fonctionnel *Canaux d'échange de message – Transmission de message*,
- une relation de dépendance de l'îlot fonctionnel *Canaux d'échange de message – Envoi de message* vers l'îlot fonctionnel *Archivage de message*.

La deuxième relation de dépendance n'est possible d'après Règle d'urbanisme – Flux et stock que si l'îlot fonctionnel *Archivage de message* était de type flux. Ce changement de type revient à considérer que la parcelle fonctionnelle *Stocker un message* ne produit qu'une date de stockage et non le message stocké.

L'hypothèse pour la suite de l'expérimentation est que l'urbaniste fonctionnel a validé la vue fonctionnelle de la zone *Messagerie* avec un alignement avec la vue métier mesuré à 7/8. L'appréciation réelle de l'urbaniste fonctionnel sur le bilan de l'alignement est reportée dans l'évaluation de la démarche DA4EA du §3.7.

3.6.1.5 Conception automatisée des quartiers fonctionnels

Le temps d'exécution du programme implémentant l'algorithme bactériologique, pour un taux de mutation variant de 0% à 100%, est représenté dans la **Figure 47**. Ce temps correspond à la vitesse de convergence de l'algorithme bactériologique. Dès que la convergence est atteinte, le temps d'exécution devient en effet négligeable.

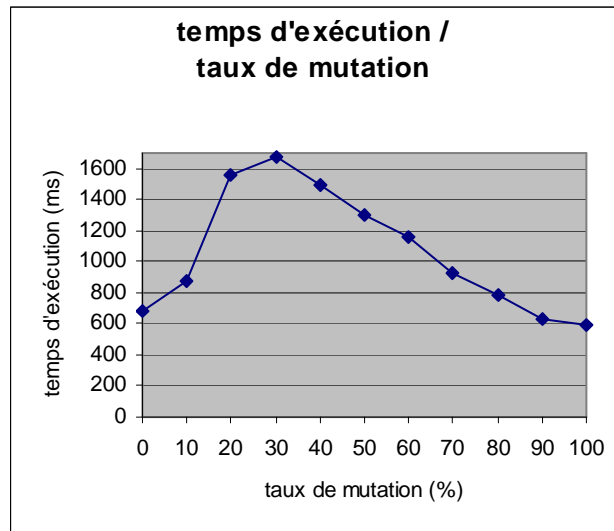


Figure 47 : Temps d'exécution de l'algorithme bactériologique par taux de mutation

La convergence la plus rapide de l'algorithme bactériologique se situe pour un taux de mutation bas (entre 0% et 10%) ou pour un taux de mutation élevé (entre 70% et 100%). Dans ces deux cas, l'algorithme bactériologique converge pour un minimum local. La mutation est soit trop lente pour un taux de mutation bas, soit trop rapide pour un taux de mutation élevé, pour atteindre de bons minima. La conséquence est que l'algorithme ne génère pas un nombre significatif d'entités fonctionnelles de qualité pour un taux de mutation faible ou un taux de mutation élevé.

Les autres résultats de l'algorithme bactériologique sont donc détaillés ci-après avec un taux de mutation variant de 5% à 50%.

Pour chaque taux de mutation, le nombre cumulé d'entités fonctionnelles résultat de séries de dix exécutions de l'algorithme, est représenté dans la **Figure 48**.

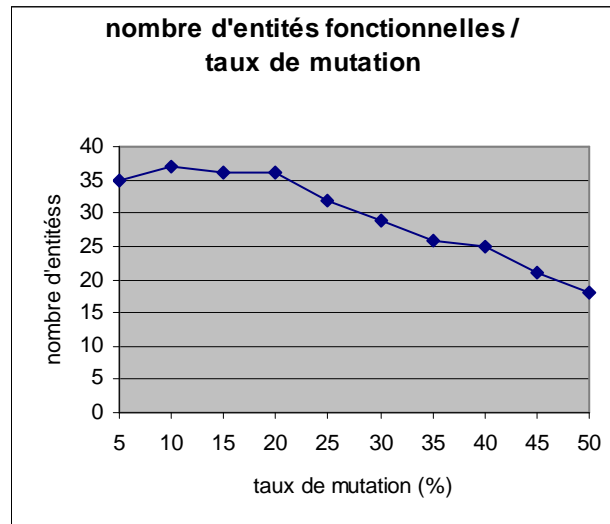


Figure 48 : Nombre d'entités fonctionnelles générées par l'algorithme bactériologique par taux de mutation

Plus le taux de mutation augmente, plus le nombre d'entités fonctionnelles générées par l'algorithme diminue. En effet, la convergence vers des entités fonctionnelles à couplage fort est favorisée par un taux de mutation élevé. La définition de la fonction bactériologique de filtrage dans le §3.5.4.1 implique que les solutions à couplage plus faible ne sont pas prises en compte.

Plus le taux de mutation est faible, plus le nombre moyen d'îlots fonctionnels par entité fonctionnelle augmente. La raison est un couplage plus faible des îlots de l'entité fonctionnelle lorsque le taux de mutation est faible. La convergence vers un nombre d'entités fonctionnelles stable est de ce fait plus lente.

Lorsque le taux de mutation est élevé, les entités fonctionnelles ayant un couplage fort entre leurs îlots sont rapidement générées. Du fait de la fonction de classement, chaque entité fonctionnelle ainsi conçue n'insère pas d'îlots fonctionnels diminuant fortement son couplage. La convergence vers un nombre d'entités fonctionnelles stable est de ce fait plus rapide.

L'allure de la courbe du temps d'exécution par rapport au taux de mutation est donc très corrélée à la courbe du nombre d'entités fonctionnelles générées.

Le nombre d'entités fonctionnelles pour un taux de mutation supérieur à 50% non représenté dans la **Figure 48** continue de décroître jusqu'à une valeur nulle pour un taux de 100%.

La qualité moyenne d'une entité fonctionnelle est représentée dans la **Figure 49** par rapport à une estimation a posteriori de son potentiel à devenir un quartier fonctionnel. Cette estimation respecte une notation de 0 à 3 à partir de l'intensité du couplage entre îlots fonctionnels d'une même entité fonctionnelle :

- 3 – entité fonctionnelle à couplage très fort,
- 2 – entité fonctionnelle à couplage fort,
- 1 – entité fonctionnelle à couplage moyen,
- 0 – entité fonctionnelle à couplage faible.

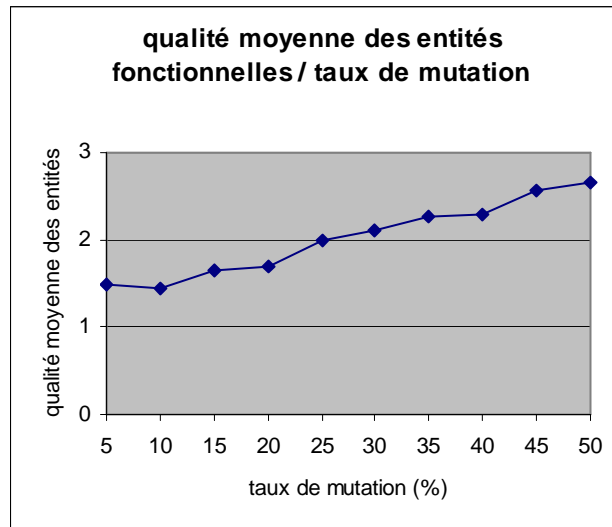


Figure 49 : Qualité moyenne des entités fonctionnelles mémorisées par taux de mutation

La qualité moyenne des entités fonctionnelles mémorisées est croissante. Le premier résultat est que, quelque soit le taux de mutation, la qualité des entités fonctionnelles mémorisées par l'algorithme bactériologique représente au minimum un couplage moyen – fort. La fonction de mémorisation des meilleures entités fonctionnelles dépendante du seuil de regroupement ainsi que la fonction de filtrage des plus mauvaises entités fonctionnelles dépendantes de la pertinence de l'entité fonctionnelle sont donc efficaces.

La qualité moyenne des entités fonctionnelles pour un taux de mutation supérieur à 50% non représenté dans la **Figure 49** se stabilise.

La qualité cumulée des entités fonctionnelles mémorisées est représentée dans la **Figure 50**.

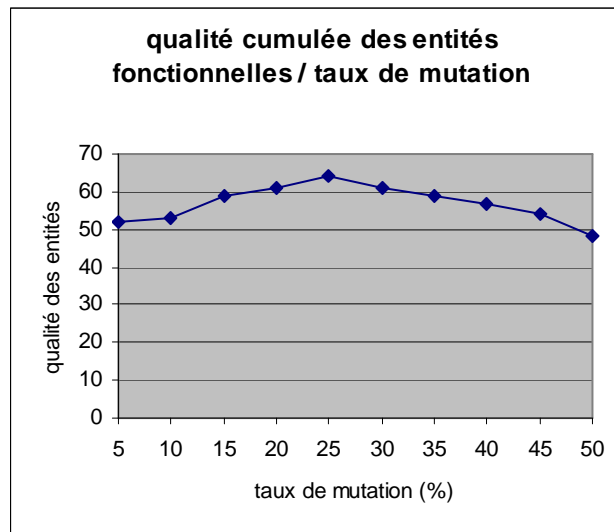


Figure 50 : Qualité cumulée des entités fonctionnelles mémorisées par taux de mutation

Par comparaison avec le nombre d'entités fonctionnelles générées, et donc non filtrées de la **Figure 49**, le maximum de qualité atteint pour les entités fonctionnelles ne correspond pas au maximum du nombre d'entités fonctionnelles. Ce maximum est atteint pour un taux de mutation de 25% pour la qualité cumulée alors que le maximum du nombre d'entités fonctionnelles générées est atteint pour un taux de mutation de 10%. Ceci signifie, que l'optimum de génération des quartiers fonctionnels est un compromis à atteindre entre ces deux courbes.

Le nombre d'entités fonctionnelles mémorisées pour un taux de mutation supérieur à 50%, non représenté dans la **Figure 50**, continue de décroître jusqu'à une valeur nulle atteinte pour un taux mutation de 100%.

Les entités fonctionnelles E les plus fréquemment générées lors de cette conception automatisée des quartiers fonctionnels sont par ordre de gain (quotient de la pertinence de l'entité par le seuil du regroupement dont l'entité est le résultat) :

1. $E = \{\text{Répondeur - Enregistreur - Envoi de message vocal, Structure de message}\}$
2. $E = \left\{ \begin{array}{l} \text{Répondeur - Enregistreur - Envoi de message vocal,} \\ \text{Répondeur - Enregistreur - Création de message vocal, Structure de message} \end{array} \right\}$
3. $E = \{\text{Canaux d'échange de message, Archivage de message}\}$
4. $E = \{\text{Canaux d'échange de message, Archivage de message, Tri de message}\}$
5. $E = \left\{ \begin{array}{l} \text{Canaux d'échange de message, Archivage de message,} \\ \text{Répondeur - Enregistreur - Envoi de message vocal} \end{array} \right\}$
6. $E = \{\text{Canaux d'échange de message, Archivage de message, Structure de message}\}$
7. $E = \left\{ \begin{array}{l} \text{Répondeur - Enregistreur - Envoi de message vocal,} \\ \text{Répondeur - Enregistreur - Création de message vocal} \end{array} \right\}$
8. $E = \left\{ \begin{array}{l} \text{Répondeur - Enregistreur - Envoi de message vocal,} \\ \text{Canaux d'échange de message} \end{array} \right\}$

3.7 Évaluation de la démarche DA4EA

Lors du prototypage de la démarche DA4EA, l'activité de validation du PLU fonctionnel entourée dans la **Figure 51** recouvre l'évaluation de la démarche. La démarche est évaluée par les urbanistes fonctionnels qui l'ont prototypée.

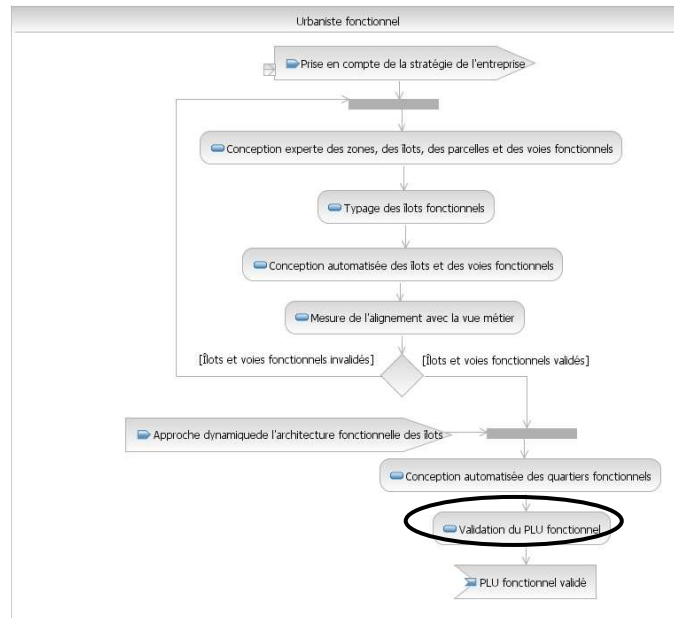


Figure 51 : Activité de validation du PLU fonctionnel dans la démarche DA4EA

Le prototypage de la démarche DA4EA a eu lieu en avril 2008 à partir de deux zones fonctionnelles :

- la zone *Messagerie*, nommée Z1 dans l'évaluation, sur laquelle la démarche dans sa totalité est expérimentée (cf. §3.6.1),
- la zone *Device*, nommée Z2 dans l'évaluation, dont l'urbaniste est Benoit Hellequin de France Télécom R&D, est dédiée aux terminaux. Seule l'activité d'alignement avec la vue métier n'est pas évaluée pour cette zone.

Dans les deux cas, l'urbaniste, expert de la zone fonctionnelle, a réalisé plusieurs incréments de la démarche DA4EA. Le document en entrée du prototypage de la démarche est la description d'une zone fonctionnelle par cet expert. La seule différence entre les deux zones est que les quartiers de la zone *Messagerie* étaient conçus au préalable, alors qu'ils ne l'étaient pas pour la zone *Device*.

Chaque itération due à l'urbaniste fonctionnel peut être liée à :

- une modification du typage des îlots fonctionnels,
- un ajout d'une ou plusieurs parcelles fonctionnelles,
- à l'ajout de voies entre îlots fonctionnels suite à la scission d'un îlot en un îlot de type flux et un îlot de type stock.

3.7.1 Contexte du prototypage de la démarche DA4EA

Le prototypage de la démarche DA4EA a été accompagné d'un support par le concepteur de la démarche. Un outillage d'activités automatisées a complété ce support. Après la mise en œuvre de la démarche DA4EA par l'urbaniste responsable de la zone fonctionnelle étudiée, un questionnaire lui a été proposé. Un bilan de la démarche a été réalisé à partir des réponses à ce questionnaire.

Le questionnaire contient dix questions rattachées à quatre thèmes :

- l'approche DA4EA, avec une évaluation de la complexité de l'approche par rapport à son intérêt pour l'urbaniste,
- le contenu architectural issu de l'application de la démarche DA4EA avec une évaluation de l'architecture obtenue et de la qualité de la conception, en particulier en termes de traçabilité,
- le coût de l'approche DA4EA comparé au coût de l'approche de conception d'une zone fonctionnelle par un expert sans automatisation d'activités,
- le bilan détaillé par activité automatisée de la démarche DA4EA.

3.7.1.1 Approche DA4EA

Q1. Trouvez-vous l'approche DA4EA "très compliquée – compliquée – simple – très simple" à mettre en œuvre ?

Q2. Trouvez-vous l'approche DA4EA "sans intérêt – pas très utile – utile – essentielle" pour votre travail d'urbaniste fonctionnel ?

Q3. Trouvez-vous le support de cette approche DA4EA "sans intérêt – pas très utile – utile – essentiel" ?

3.7.1.2 Urbanisme et approche DA4EA

Q4. L'architecture de votre zone fonctionnelle conçue avec DA4EA est-elle "vraiment pire – pire – meilleure – vraiment meilleure" que l'architecture de la zone fonctionnelle précédente ?

Q5. La qualité de la conception de votre zone fonctionnelle conçue avec DA4EA est-elle "vraiment pire – pire – meilleure – vraiment meilleure" que la qualité de la zone fonctionnelle précédente ?

3.7.1.3 Coût de l'approche DA4EA

Q6. Le coût de la conception d'une zone fonctionnelle avec DA4EA est-il "vraiment plus important – plus important – moins important – vraiment moins important" que le coût de conception précédent ?

3.7.1.4 Automatisation et approche DA4EA

Q7. La définition des îlots, et des voies entre îlots, de votre zone fonctionnelle avec la démarche DA4EA est-elle "vraiment pire – pire – meilleure – vraiment meilleure" que la définition actuelle ?

Q8. Trouvez-vous la mesure de l'alignement de la vue fonctionnelle avec la vue métier proposée par la démarche DA4EA "sans intérêt – pas très utile – utile – essentielle" ?

Q9. Trouvez-vous le bilan faisant suite à la mesure d'alignement de la vue fonctionnelle avec la vue métier proposée par la démarche DA4EA "sans intérêt – pas très utile – utile – essentiel" pour votre travail d'urbaniste fonctionnel ?

Q10. Trouvez-vous les résultats de la conception bactériologique des quartiers fonctionnels proposée par la démarche DA4EA "vraiment pire – pire – meilleure – vraiment meilleure" que les quartiers précédents ?

3.7.2 Bilan du prototypage de la démarche DA4EA

Les réponses de chaque urbaniste responsable de zone fonctionnelle (Z1 à Z2) sont reportées dans le Tableau 9 de la façon suivante :

- Q1 : approche très compliquée "—", compliquée "-", simple "+", très simple "++",
- Q2 : approche sans intérêt "—", peu utile "-", utile "+", essentielle "++",
- Q3 : support sans intérêt "—", peu utile "-", utile "+", essentiel "++",
- Q4 : architecture de la zone vraiment pire "—", pire "-", meilleure "+", vraiment meilleure "++",
- Q5 : qualité de la conception vraiment pire "—", pire "-", meilleure "+", vraiment meilleure "++",
- Q6 : coût vraiment plus important "—", plus important "-", moins important "+", vraiment moins important "++",
- Q7 : définition des îlots et des voies vraiment pire "—", pire "-", meilleure "+", vraiment meilleure "++",
- Q8 : mesure d'alignement avec la vue métier sans intérêt "—", peu utile "-", utile "+", essentielle "++",
- Q9 : bilan de l'alignement avec la vue métier sans intérêt "—", peu utile "-", utile "+", essentiel "++",
- Q10 : conception des quartiers vraiment pire "—", pire "-", meilleure "+", vraiment meilleure "++".

Zone fonctionnelle	Approche			Urbanisme		Coût	Automatisation			
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Z1	+	++	++	+	+	+	+			+
Z2	+	++	++	+	+	+	+	++	++	+

Tableau 9 : Réponses au questionnaire du bilan de la démarche DA4EA.

3.7.3 Analyse du bilan du prototypage de la démarche DA4EA

Même si le nombre de zones conçues lors du prototypage de la démarche DA4EA est faible, de premiers enseignements peuvent être tirés (cf. **Figure 52**) :

- le retour des deux responsables de zone fonctionnelle est globalement positif,
- l'approche DA4EA est très intéressante et simple,
- les activités automatisées ont de bons retours quant à la qualité des résultats,
- parmi ces activités automatisées, l'alignement de la zone fonctionnelle avec la vue métier et son évaluation ont eu un très bon retour de l'urbaniste qui a pu les mettre en œuvre,
- le coût est moins important que celui des anciennes approches du fait de l'outillage d'activités automatisées.

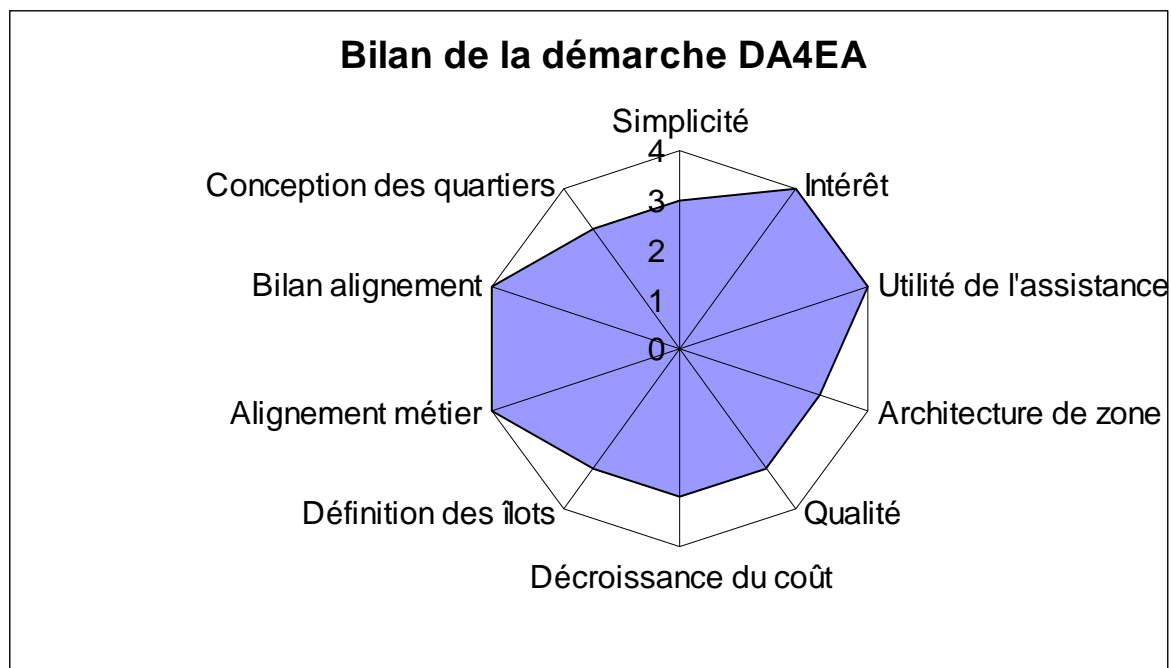


Figure 52 : Bilan de la démarche DA4EA

En complément de l'automatisation outillée de certaines activités, la nécessité d'une formation adressée aux urbanistes fonctionnels est apparue lors du bilan.

3.7.4 Bilan comparatif avec la zone Messagerie existante

En préalable de l'évaluation de la démarche par l'urbaniste fonctionnel (cf. §3.7), un bilan comparatif est dressé entre la zone fonctionnelle *Messagerie* conçue avec la démarche DA4EA et la zone *Messagerie* existante. Ce bilan contient des éléments chiffrés de comparaison entre l'architecture fonctionnelle de la zone *Messagerie* conçue totalement par des experts et l'architecture fonctionnelle de la zone avec la démarche DA4EA.

Le bilan est fait pour les concepts du PLU fonctionnel impactés par la démarche DA4EA, soient, dans l'ordre de la démarche, l'îlot, la voie et le quartier. Le Tableau 10 représente l'alignement de chacun de ces concepts entre la vue fonctionnelle entièrement conçue par l'urbaniste fonctionnel expert de la zone *Messagerie*, et la vue fonctionnelle générée par la démarche DA4EA.

		Zone <i>Messagerie</i> DA4EA / Zone <i>Messagerie</i> "expert"
Îlot	Aligné	6
	Non aligné	2
Voie	Alignée	5
	Non alignée	4
Quartier	Aligné	0
	Non aligné	2

Tableau 10 : Alignement de la zone *Messagerie* "expert" et de la zone *Messagerie* DA4EA.

Le non alignement des îlots fonctionnels et des voies fonctionnelles est dû à la scission de l'îlot fonctionnel multi-typé *Répondeur – Enregistreur*. Néanmoins, les voies de la zone *Messagerie* conçue uniquement par l'expert sont reportées sur au moins un des îlots fonctionnels issus de la scission :

- la voie de l'îlot *Répondeur – Enregistreur* vers l'îlot *Canaux d'échanges de message* est reportée sur l'îlot *Répondeur – Enregistreur – Envoi de message vocal*,
- la voie de l'îlot *Répondeur – Enregistreur* vers l'îlot *Structure de message* est reportée sur l'îlot *Répondeur – Enregistreur – Envoi de message vocal* et sur l'îlot *Répondeur – Enregistreur – Création de message vocal*.

Aucune des entités fonctionnelles générées par l'algorithme bactériologique ne correspond à un des quartiers conçus par l'urbaniste fonctionnel de la zone *Messagerie*. Comme pour l'alignement entre la vue fonctionnelle de la zone et le processus métier supporté, l'évaluation par l'urbaniste de ces quartiers non alignés est reportée dans le §3.7.

3.8 Conclusion

La synthèse par thèmes du bilan de la démarche DA4EA (cf. §3.7) est reportée dans la **Figure 53** avec :

- l'intérêt et la pertinence de l'approche,
- l'intérêt de l'automatisation pour assister l'urbaniste fonctionnel lors de la conception de sa zone fonctionnelle,
- l'apport de la démarche DA4EA lors de la conception d'une zone fonctionnelle,
- un coût moins important que celui des approches précédentes.

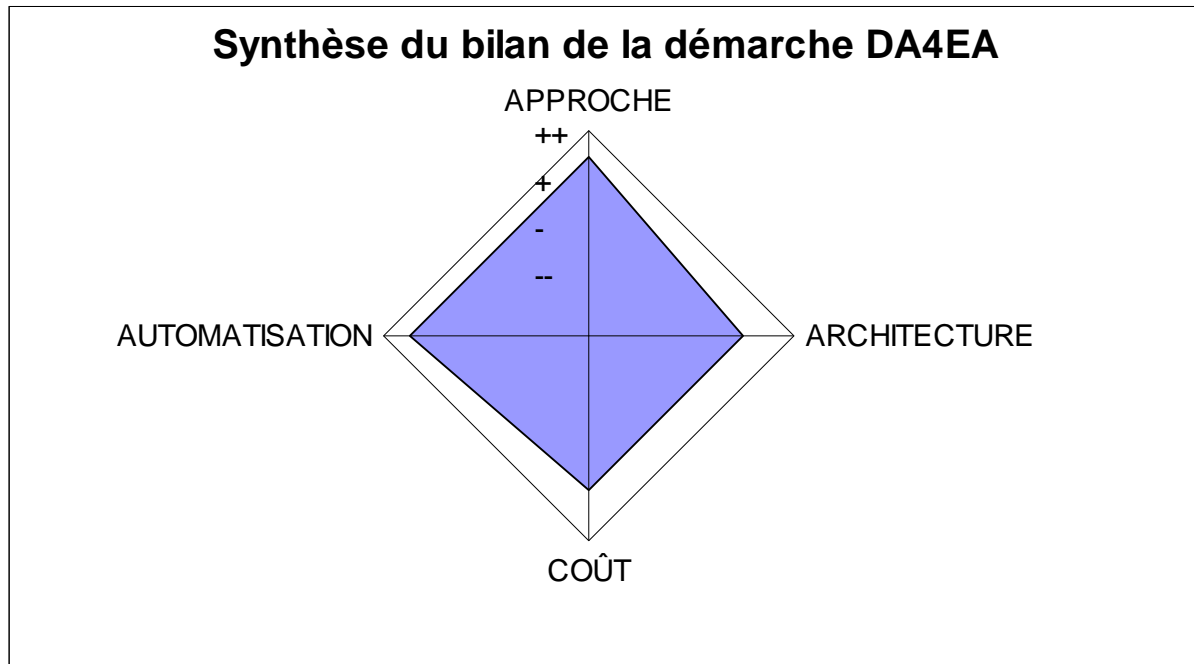


Figure 53 : Synthèse du bilan de la démarche DA4EA

Le bon résultat du bilan, malgré le peu d'expérimentations faites avec la démarche DA4EA, est instructif sur l'apport d'une automatisation de certaines activités avec l'ingénierie des modèles. Cette automatisation permet d'offrir un coût de conception moins important et de rendre plus simple l'approche pour l'urbaniste utilisateur de la démarche.

4. La démarche EA4UP de développement de système assisté par l'urbanisme fonctionnel d'un SI

4.1 Principes

En reprenant l'illustration du logement (cf. §2.3.2), le problème à résoudre est la détermination, pour tous logements à bâtir, d'une méthode de construction tenant compte du logement modèle conçu fonctionnellement avec la démarche DA4EA. Cette démarche qui s'inscrit dans le processus de développement EUP [Ambler, 2005] doit faciliter la prise en compte de l'EA par les développeurs de systèmes. C'est l'objectif de la démarche EA4UP qui a la propriété d'intégrer un PLU fonctionnel [Simonin, 2008].

4.1.1 Pourquoi l'urbanisme fonctionnel ?

Un SI devient durable grâce, en particulier, à la réutilisation de ses systèmes. Cette réutilisation est favorisée par la description partagée de la vue fonctionnelle de chacun de ces systèmes. La description partagée s'appuie sur la vue fonctionnelle du SI telle que conçue avec la démarche DA4EA. Les liens entre vues sont inspirés du cycle de développement en Y tel que celui repris dans le processus de développement 2TUP (2 Track Unified Process) [Roques, 2004]. Le cycle en Y est tel que la vue applicative d'un système réalise l'analyse fonctionnelle de ce système et est déployée sur son architecture technique. L'insertion de la vue fonctionnelle du SI et de la vue métier de l'entreprise nécessite de compléter ce cycle. Les relations entre la vue fonctionnelle d'un système et les autres vues du cadre de l'urbanisme sont représentées dans le diagramme de classes de la **Figure 54**.

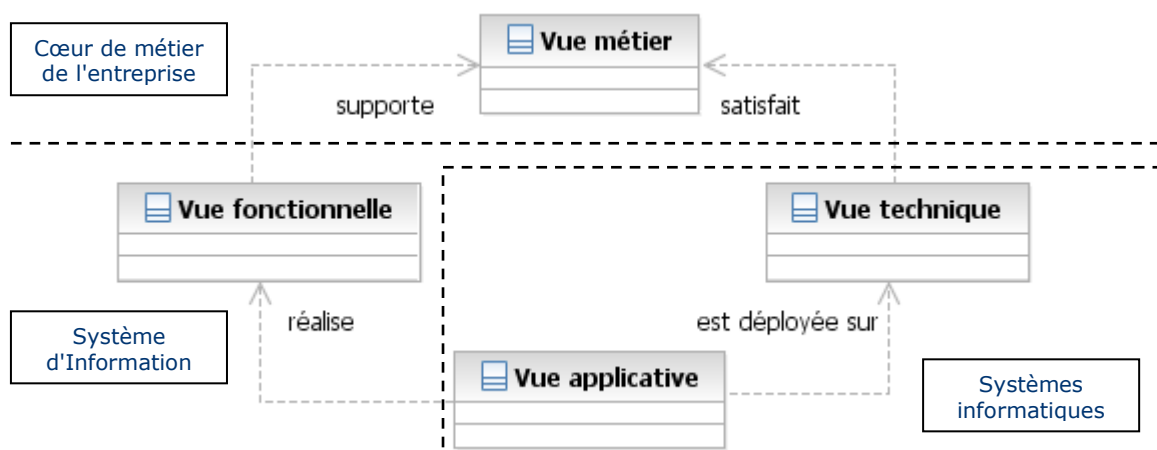


Figure 54 : Vues du cadre de l'urbanisme, cœur de métier de l'entreprise, SI et systèmes

La vue fonctionnelle du SI supporte la vue métier de l'entreprise. Par rapport aux concepts de ces deux vues définis dans le §2.2.1.2 et dans le §2.2.1.3, ceci signifie que les îlots fonctionnels et leurs parcelles supportent chaque procédure métier de l'entreprise. L'alignement entre la vue fonctionnelle et la vue métier défini dans le §3.4 illustre cette notion de support d'une vue par l'autre.

La vue technique, qui est propre aux systèmes informatiques, satisfait la vue métier de l'entreprise. Dans le contexte d'une entreprise telle que France Télécom, les nœuds d'exécution et les protocoles définis dans le §2.2.1.2 satisfont les procédures métier liées aux réseaux nouvelles générations (NGN : Next-Generation Networks) [Knightson, 2005] tels que ceux respectant l'environnement architectural IMS (IP Multimedia Subsystem) [Cuevas, 2006].

La vue applicative, propre aux systèmes informatiques comme la vue technique, représente la réalisation de la vue fonctionnelle des systèmes déployés sur la vue technique. Toutes les parcelles fonctionnelles du SI ne sont pas obligatoirement réalisées par un élément d'un système informatique. C'est par exemple le cas de la réalisation par un acteur humain d'une parcelle fonctionnelle de transmission d'un bon de commande saisi en front-office pour traitement en back-office.

Chacune de ces vues possède un aspect dynamique (cf. §2.3.2). Pour chaque vue, il existe en effet un lien avec des interactions réalisant l'usage d'un système. Pour la vue fonctionnelle, l'aspect dynamique peut être modélisé par des diagrammes de séquence UML représentant les interactions entre îlots fonctionnels instanciés lors du déroulement d'un scénario de cas d'utilisation d'un système (cf. §3.2). Il en est de même pour la vue applicative où des interfaces applicatives de type requête ou de type réponse sont définies afin de permettre des interactions entre instances de composants applicatifs.

Définition : Une interface applicative représentant une interaction entre instances de composants applicatifs est :

- de type requête si l'interaction est une demande ou si l'interaction correspond à l'émission de paramètres d'entrée d'une requête,
- de type réponse si l'interaction est une réponse à une demande ou si l'interaction correspond à l'émission de paramètres de sortie d'une requête.

4.1.2 Pourquoi une approche dynamique ?

L'architecture SOA nécessite de considérer l'aspect dynamique de chaque vue d'architecture. Chaque procédure de la vue métier est en effet supportée par une séquence de services. Dans la démarche EA4UP, la séquence de services est représentée dans la vue fonctionnelle du système par une séquence d'interactions entre un scénario et un îlot fonctionnel. Chaque interaction est du type requête ou du type réponse. Elle correspond à une parcelle fonctionnelle instanciée lors de l'interaction. La séquence d'interactions entre un scénario et un îlot fonctionnel est réalisée dans la vue applicative par une séquence d'interactions entre le scénario et les composants applicatifs. Ces interactions correspondent à l'utilisation des interfaces applicatives de ces composants.

La définition des interactions signifie la réutilisation des îlots fonctionnels ou celle des composants applicatifs. L'utilisation d'un îlot fonctionnel induit en effet un accès à une donnée par l'intermédiaire d'une de ses parcelles fonctionnelles. Cet accès induit une interaction avec cet îlot fonctionnel. Il en est de même pour un composant applicatif et son interface. L'interface applicative offre en effet un accès à ce composant durant une interaction réalisant un cas d'utilisation d'un système.

La relation de dépendance pour les concepts de la vue applicative est définie de façon analogue à la relation de dépendance pour les concepts de la vue fonctionnelle (cf. §3.4.1 et §3.5.2).

Définition :

- deux interfaces applicatives sont en relation de dépendance si toutes deux sont exclusivement du type requête, ou du type réponse, et si l'une d'entre elles succède à l'autre dans un diagramme de séquence UML représentant un scénario. Une interface applicative IA_1 succède à une interface applicative IA_2 si la ligne de vie cible de IA_2 est la ligne de vie à l'origine de IA_1 ;
- deux composants applicatifs sont en relation de dépendance s'il existe une interaction de type requête de l'un vers l'autre dans un diagramme de séquence UML. Le composant à l'origine de l'interaction dépend du composant à la cible de cette interaction.

Le principe est d'axer la conception de la vue fonctionnelle et la conception de la vue applicative sur l'aspect dynamique de l'usage du système. L'architecte du système doit en effet déduire la conception d'une relation de dépendance entre deux îlots fonctionnels, ou une relation de dépendance entre deux composants applicatifs, à partir du déroulement d'un scénario. D'après la Règle d'urbanisme – Interactions et voies du §3.2, la relation de dépendance se traduit par une interaction entre les instances de ces îlots fonctionnels ou entre les instances de ces composants applicatifs.

La Sémantique UML – Approche dynamique et diagramme de séquence (cf. §2.1.5) est illustrée dans le cas des îlots fonctionnels par la **Figure 55**.

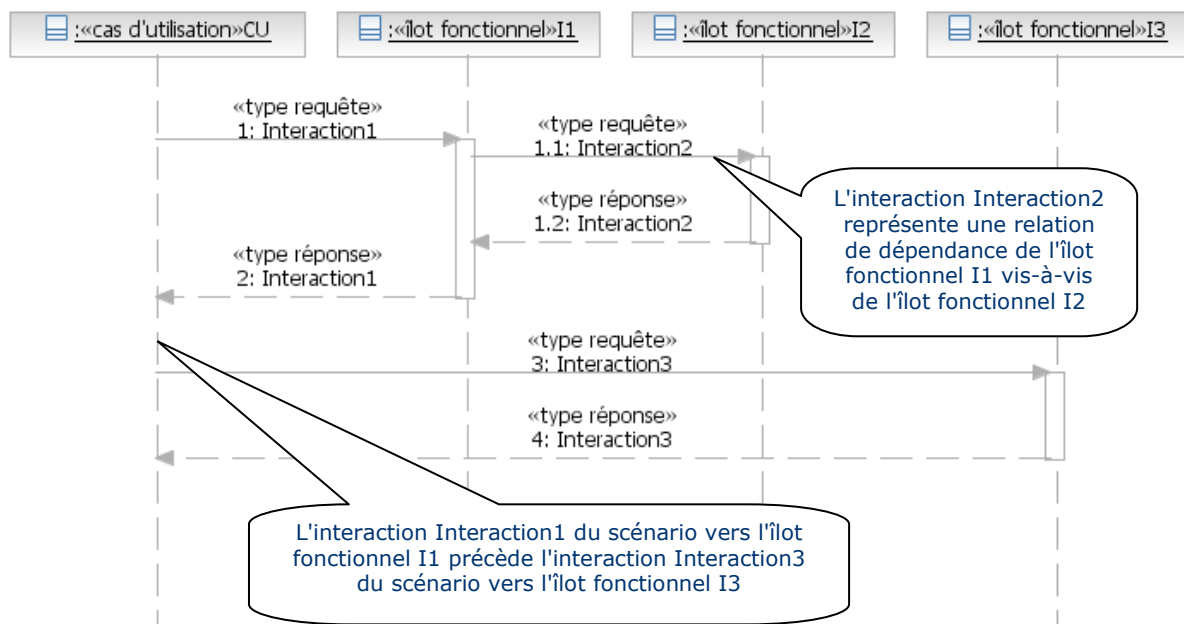


Figure 55 : Ordonnancement temporel et relation de dépendance dans un diagramme de séquence

De même, il existe une cohérence des relations de dépendance entre parcelles fonctionnelles, ou entre composants applicatifs, avec les relations de dépendance entre les données produites, respectivement, par ces parcelles ou par ces composants. La Règle d'urbanisme – Relation de dépendance entre données précise les conditions de dépendance au sens UML entre deux données logiques de la vue fonctionnelle ou deux données physiques de la vue applicative.

Règle d'urbanisme – Relation de dépendance entre données n°6

- une relation de dépendance entre parcelles fonctionnelles de type requête implique une relation de dépendance au sens UML entre les données logiques produites par ces parcelles fonctionnelles ;
- une donnée logique DL_1 produite par une parcelle fonctionnelle de type requête dépend a priori d'une donnée logique DL_2 utilisée par la parcelle. La condition est la relation de dépendance de la parcelle fonctionnelle produisant la donnée logique DL_1 vis-à-vis de la parcelle fonctionnelle produisant la donnée logique DL_2 ,
- une relation de dépendance entre interfaces applicatives de type requête implique une relation de dépendance au sens UML entre les données physiques produites par ces interfaces applicatives ;
- une donnée physique DP_1 fournie par une interface applicative de type requête dépend a priori d'une donnée physique DP_2 utilisée par l'interface. La condition est la relation de dépendance de l'interface applicative de type requête produisant la donnée physique DP_1 vis-à-vis de l'interface applicative de type requête produisant la donnée physique DP_2 .

4.1.3 Démarche EA4UP et ingénierie des modèles

La durabilité d'un SI est liée à la réutilisation des éléments de ce SI. Le choix est fait dans la démarche EA4UP de souligner cette réutilisation dès la conception de la vue fonctionnelle et durant la conception de la vue technique du système. Un élément réutilisable possède en effet une vue applicative déjà conçue. Cette vue applicative réalise une vue fonctionnelle de l'élément et est déployée sur sa vue technique. Afin de cadrer au mieux la réutilisation d'un élément du SI lors de la conception de l'architecture d'un système, une dépendance supplémentaire de la vue technique vis-à-vis de la vue fonctionnelle est ajoutée (cf. **Figure 56**). Les vues d'architecture d'un système sont donc toutes interdépendantes.

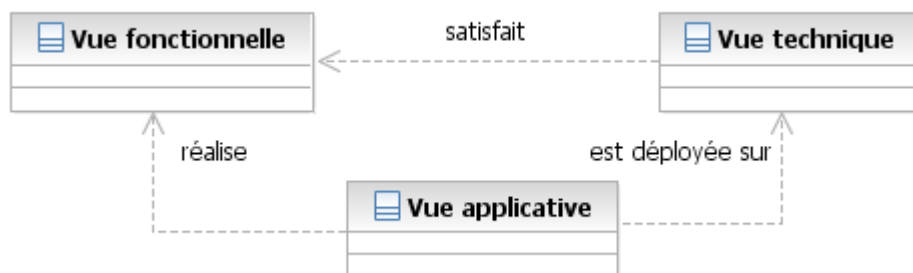


Figure 56 : Vues d'architecture d'un système et démarche DA4EA

Du fait de ces interdépendances, la plupart des activités sont assimilables à des transformations de modèles telles que décrites dans l'approche MDE ou dans l'approche MDA (cf. §2.3.3). Par exemple, un modèle de la vue technique est directement ou indirectement issu d'une transformation de modèles contrainte par un modèle de la vue fonctionnelle. C'est le cas de l'analyse technique où les exigences non fonctionnelles sont contraintes par l'analyse fonctionnelle du système. Chaque exigence non fonctionnelle est en effet associée à un ou plusieurs éléments issus de l'analyse du système. Ces transformations de modèle sont réalisées dans la démarche EA4UP par des experts ou sont automatisées. Dans le premier cas, la transformation est une conception assistée d'un contrôle du respect de règles d'urbanisme (cf. §2.2.1.6). Dans le second cas, l'automatisation nécessite en amont un codage des éléments du PIM par des éléments du PDM. Le PIM est un modèle produit lors du développement du système et le PDM est un modèle de référence conçu par les urbanistes. Le codage est une association, réalisée par un développeur du système, de chaque élément du PIM avec un ou plusieurs éléments du PDM. Cette association est en générale représentée dans le méta-modèle des vues d'urbanisme par une association entre le ou les concepts instanciés dans le PIM et le ou les concepts instanciés dans le PDM associé.

Le modèle de référence pour la vue fonctionnelle est le PLU fonctionnel (cf. §4.1.1). Pour la vue technique, le modèle de référence est un ensemble de préconisations de solutions techniques ou de vue technique d'éléments réutilisables du SI. La vue applicative de ces éléments constitue le modèle de référence de la vue applicative. Le rôle de l'expert dans le cas d'une automatisation de l'activité est de concevoir une vue en s'aidant des solutions produites de façon automatique.

Définition : Lors d'une transformation de modèle contrainte par un PDM, une activité de codage consiste à associer un ou plusieurs éléments du PDM à chaque élément du PIM. Le résultat du codage permet la transformation de modèle grâce à l'application de règles d'urbanisme.

Une illustration du codage est la transformation d'un modèle d'entités en un modèle de données logiques de la vue fonctionnelle. L'activité de codage est ici l'association de chaque entité à un ou plusieurs îlots fonctionnels produisant les données extraites de l'entité [Simonin, 2007]. La règle d'urbanisme contraignant la transformation est la Règle d'urbanisme – Donnée produite et îlot fonctionnel du §2.2.1.2.

Dans cette approche d'ingénierie des modèles, plusieurs sortes de transformation de modèle sont mises en œuvre dans la démarche EA4UP. La démarche EA4UP peut être résumée à un enchaînement de transformations de modèles réalisées par des experts ou automatisées.

La démarche EA4UP est fondée sur des transformations de modèle à partir de la définition d'un PIM, d'un PSM et parfois d'un PDM. Le PDM est dans notre approche une extension de la notion de PDM définie par l'OMG. Dans EA4UP, l'originalité de l'utilisation de l'approche MDA est d'assimiler les modèles conçus par les urbanistes fonctionnels, les modèles conçus par les urbanistes techniques ou les modèles conçus par les urbanistes applicatifs à des PDM. La raison est que ce rôle de modèle définissant des contraintes lors d'une transformation de modèles joué par le PDM prend tout son sens avec les modèles d'EA d'un SI, en particulier d'un SI durable. Tous les modèles d'architecture résultant du développement d'un système doivent en effet respecter au mieux les modèles d'EA les concernant :

- le PLU fonctionnel d'un SI, tel que conçu avec la démarche DA4UP, contraint la vue fonctionnelle d'un système du SI,
- l'ensemble de préconisations, soit purement techniques, soit définies par la vue technique d'éléments à réutiliser du SI, contraint la vue technique du système à développer,
- les composants applicatifs réutilisables et leurs interfaces contraignent la conception de l'architecture applicative d'un système.

Chaque activité de la démarche EA4UP, en dehors de celles composant la macro-activité de collecte des exigences, possède au moins une activité en amont. Ces activités nécessitent donc en entrée le résultat d'une activité amont représentée par un PIM dans l'activité en cours. De plus, elles peuvent être contraintes par le résultat d'une autre activité amont représentée par un PSM, résultat de cette activité. La démarche EA4UP consiste alors en un enchaînement de transformations de modèles tel qu'illustré dans la **Figure 57**.

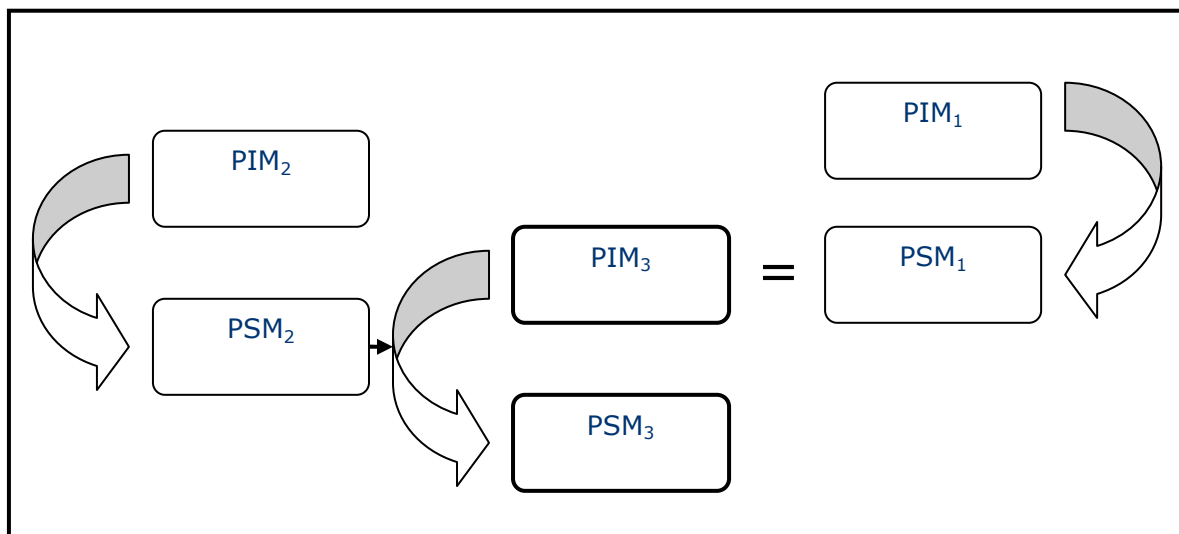


Figure 57 : Enchaînement de transformations de modèle

Dans cette illustration :

- le modèle PSM_1 résulte d'une transformation du modèle PIM_1 dans une activité amont,
- le modèle PSM_2 résulte d'une transformation du modèle PIM_2 dans une activité amont,
- le modèle PSM_3 résulte d'une transformation du modèle PIM_3 dans une activité en cours,
- dans cette activité en cours, le modèle en entrée de la transformation PIM_3 est égal au modèle PSM_1 ,
- toujours, dans l'activité en cours, le modèle PSM_2 contraint la transformation du modèle PIM_3 vers le modèle PSM_3 .

Les transformations liées aux activités des macro-activités d'architecture sont de plus contraintes par un PDM représentant le cadre de l'urbanisme à appliquer. Chacune de ces transformations peut être automatisable ou sous la responsabilité complète d'un expert. La responsabilité de l'expert est en particulier nécessaire lors d'un codage d'éléments du PIM par des éléments du PDM.

La **Figure 58** illustre le cas où la transformation décrite dans la **Figure 57** est telle que :

- le modèle PIM_1 est transformé en un modèle PSM_1 par un expert,
- le modèle PIM_2 est transformé en un modèle PSM_2 avec la contrainte du modèle PSM_1 et d'un PDM,
- La transformation du modèle PIM_2 en un modèle PSM_2 est automatique.

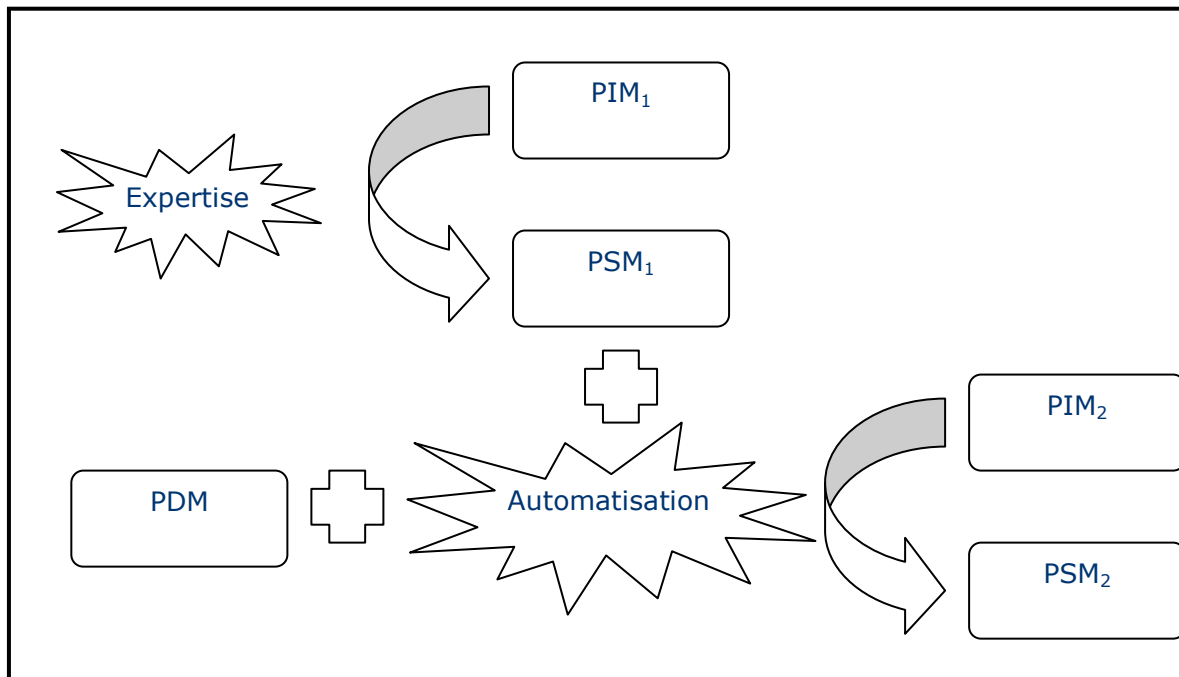


Figure 58 : Enchaînement de transformations de modèle, d'expert ou automatisable, dont l'une est conforme à un PDM

La succession d'une activité d'expert pour une approche statique par une activité automatisée pour une approche dynamique peut être représentée par un enchaînement de transformations de modèles (cf. §2.3.3).

Plus précisément, une activité dynamique nécessite dans la démarche EA4UP le résultat en entrée de l'aspect statique de cette activité. L'étude des interactions entre éléments nécessite en effet au préalable la définition de ces éléments. Cette activité statique est en fait une activité de codage d'éléments du PIM par un expert (cf. §4.1.3).

Dans l'enchaînement de transformations de modèle précédent, les deux transformations de modèles peuvent être contraintes par deux PDM représentant deux modèles différents des préconisations des urbanistes. Un enchaînement de transformations de modèles utilisé pour les activités d'architecture de la démarche EA4UP illustre dans la **Figure 59** la succession d'une activité statique par une activité dynamique telles que ces deux activités soient contraintes par l'EA.

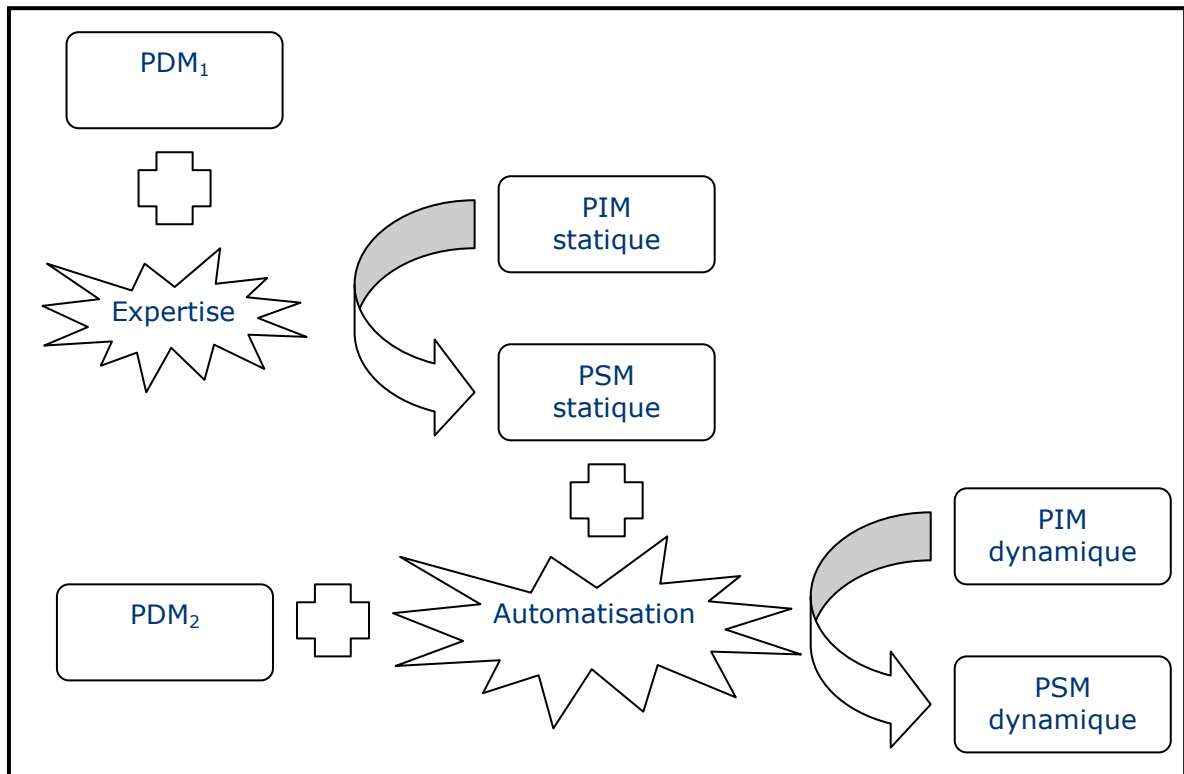


Figure 59 : Codage d'expert et transformation automatique

L'enchaînement de transformations de modèle est le suivant :

- La 1^o transformation est un codage du PIM statique avec les éléments du PDM. Ce codage réalisé par un expert permet d'obtenir un PSM statique en respectant une contrainte décrite par des règles d'urbanisme,
- la 2^o transformation est une transformation automatique du PIM dynamique en un PSM dynamique avec une contrainte implémentée à partir de règles d'urbanisme et par un PDM.

Une illustration de ce principe est l'activité de conception automatisée des îlots et des voies fonctionnels du §3.3. Le codage est celui des îlots fonctionnels par un type flux ou un type stock. Dans ce cas, la 1^o transformation permet d'associer un ou plusieurs types décrits dans PDM₁ à chaque îlot fonctionnel du PIM statique. La seconde transformation automatisée est la mise en œuvre de la Règle d'urbanisme – Scission îlot multi-typé. Permettant de définir les voies du modèle dynamique des îlots fonctionnels. Le modèle PDM₂ est défini par la Règle d'urbanisme – Flux et stock (cf. **Figure 27**).

4.1.4 Description d'ensemble de la démarche EA4UP

La démarche EA4UP recouvre les macro-activités du processus UP et leurs dérivées sur les vues d'architecture d'un système :

- La collecte auprès du client des exigences fonctionnelles et des exigences non fonctionnelles décrivant le système à développer ;
- L'analyse des exigences fonctionnelles et des exigences non fonctionnelles du système ;
- La conception de l'architecture fonctionnelle et de l'architecture technique du système,
- La conception de l'architecture applicative du système,
- L'implémentation du système,
- Les tests fonctionnels, les tests de performance et de robustesse du système.

Deux collaborateurs contribuent à la démarche EA4UP décrite dans la **Figure 60** :

- la MOA (maîtrise d'ouvrage), c'est-à-dire le client du système à développer,
- la MOE (maîtrise d'œuvre), soit l'équipe de développement du système.

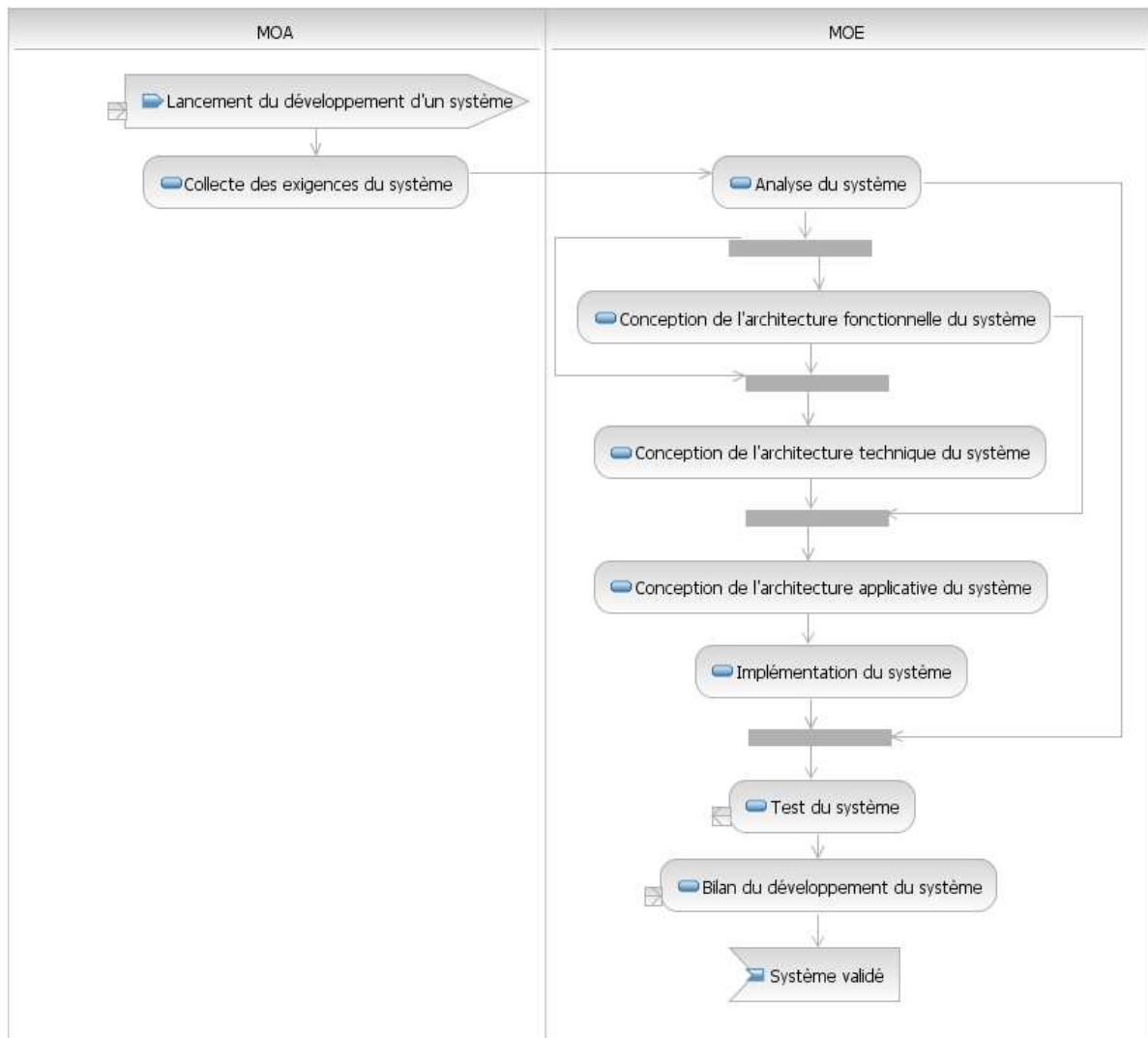


Figure 60 : Macro-activités de la démarche EA4UP

Les enchaînements des macro-activités de conception d'une architecture sont conformes aux interdépendances des vues d'architecture représentée dans la **Figure 56**.

Une activité est extraite d'une macro-activité dans le Tableau 11 par rapport à :

- sa vue, i.e. fonctionnelle, technique ou applicative,
- son aspect, i.e. statique ou dynamique,
- les concepts (cf. §2.2.1.2) produits.

Macro-activité	Activité	Vue	Statique / Dynamique	Concepts produits
Collecte des exigences	Collecte des exigences fonctionnelles statiques	Fonctionnelle	Statique	Cas d'utilisation
	Collecte des exigences fonctionnelles dynamiques		Dynamique	Scénario et Tâche
	Collecte des exigences non fonctionnelles	Technique	Statique et Dynamique	Exigence non fonctionnelle
Analyse	Analyse fonctionnelle statique	Fonctionnelle	Statique	Entité, Attribut d'entité et Relation entre entités
	Analyse fonctionnelle dynamique		Dynamique	Interaction entre entités
	Analyse technique statique	Technique	Statique	Propriété technique statique
	Analyse technique dynamique		Dynamique	Propriété technique dynamique
Conception de l'architecture fonctionnelle	Architecture fonctionnelle statique	Fonctionnelle	Statique	Îlot fonctionnel et Parcelle fonctionnelle
	Architecture fonctionnelle dynamique des îlots		Dynamique	Interaction entre îlots fonctionnels et Voie fonctionnelle
	Architecture fonctionnelle dynamique des données		Dynamique	Donnée logique et Attribut de donnée logique
Conception de l'architecture technique	Architecture technique statique	Technique	Statique	Nœud d'exécution
	Architecture technique dynamique		Dynamique	Protocole
	Architecture technique détaillée		Statique	

Conception de l'architecture applicative	Architecture applicative statique	Applicative	Statique	Composant applicatif
	Architecture applicative dynamique des composants		Dynamique	Interaction entre composants applicatifs et Interface applicative
	Architecture applicative dynamique des données		Dynamique	Donnée physique et Attribut de donnée physique
	Architecture applicative détaillée		Statique	
Implémentation	Implémentation	Applicative	Statique	
Test	Test fonctionnel	Applicative	Dynamique	
	Test de robustesse et de performance		Dynamique	

Tableau 11 : Macro-activité UP et activités EA4UP.

Chaque activité de la démarche EA4UP est détaillée dans le §4.2. Les activités spécifiques à la démarche EA4UP sont illustrées dans ce même paragraphe avec un cas d'utilisation de service télécom.

4.2 Démarche EA4UP fondée sur l'approche dynamique et l'ingénierie des modèles

Chaque activité de la démarche EA4UP est décrite par :

- son objectif,
- la ou les activités en amont de cette activité,
- la règle d'urbanisme à respecter,
- un schéma de la transformation de modèle avec son caractère automatisable ou non (cf. §4.1.3) et les concepts utiles,
- un mode de représentation des concepts produits par l'activité (cf. §4.1.4).

L'illustration de la démarche EA4UP est celle d'un service télécom de messagerie électronique. L'usage étudié est celui de la réception d'un message électronique. Le PLU fonctionnel est celui décrit dans la **Figure 31** qui sert d'illustration à la démarche DA4EA.

4.2.1 Collecte des exigences

La collecte des exigences fonctionnelles et des exigences non fonctionnelles du système à développer est une macro-activité sous la responsabilité du client du système développé, soit la MOA du système. L'utilisation fonctionnelle, ou non fonctionnelle, du système est détaillée dans cette macro-activité entourée dans la **Figure 61**.

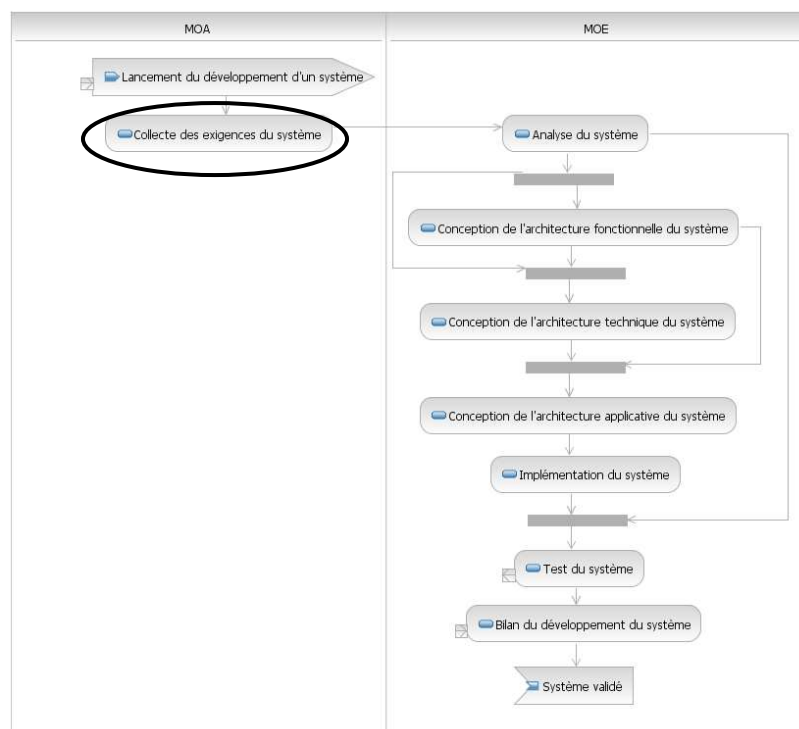


Figure 61 : Macro-activité de collecte des exigences dans la démarche EA4UP

4.2.1.1 Collecte des exigences fonctionnelles statiques

L'activité de collecte des exigences fonctionnelles statiques, entourée dans la **Figure 62**, a pour objectif de détailler les cas d'utilisation du système.

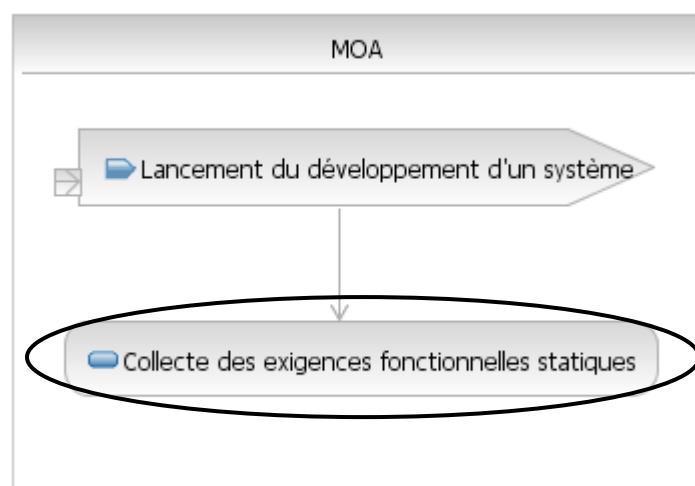


Figure 62 : Activité de collecte des exigences fonctionnelles statiques

Le concept de la vue fonctionnelle produit par cette activité est le cas d'utilisation. Le besoin de la MOA du système est décrit dans un cahier des charges. L'objectif de la collecte des exigences fonctionnelles statiques est de traduire ce besoin en cas d'utilisation du système par un de ses utilisateurs. Chaque cas d'utilisation du système détaille une série d'échanges entre l'utilisateur du système et le système offrant une valeur ajoutée à cet utilisateur.

Un mode de représentation des cas d'utilisation est une description textuelle.

Le cas d'utilisation *CU-Réception de message avec filtrage*, à l'origine de l'illustration de la démarche EA4UP, est le suivant : *L'utilisateur de la messagerie électronique reçoit un message. Il peut stocker le message dans sa boîte de messagerie uniquement si le filtrage ciblant l'objet du message a un résultat négatif.*

4.2.1.2 Collecte des exigences fonctionnelles dynamiques

L'activité de collecte des exigences fonctionnelles dynamiques, entourée dans la **Figure 63**, recouvre la description des scénarios illustrant chaque cas d'utilisation collecté lors de l'activité liée aux exigences fonctionnelles statiques (cf. 4.2.1.1).

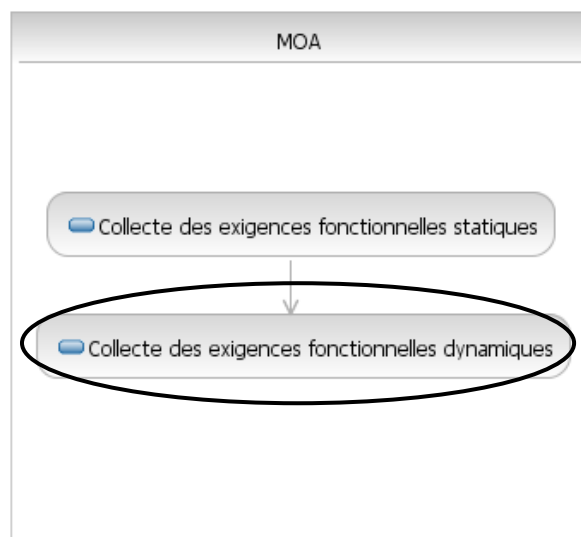


Figure 63 : Activité de collecte des exigences fonctionnelles dynamiques

Le concept de la vue fonctionnelle produit par cette activité est le scénario. Une transformation simple décrite dans la **Figure 64** est la création d'un ensemble de scénarios à partir de chaque cas d'utilisation du système. Chaque scénario représente un ensemble de résultats spécifiques aux branchements conditionnels d'un cas d'utilisation. Le rôle de l'expert est de découper chaque scénario en tâche.

La règle d'urbanisme à mettre en œuvre lors de cette transformation décrit le lien entre les cas d'utilisation collectés précédemment et les scénarios.

Règle d'urbanisme – Collecte des exigences fonctionnelles dynamiques n°7

Un scénario est une illustration d'un cas d'utilisation prenant en compte des résultats pour chaque branchement conditionnel du cas d'utilisation.

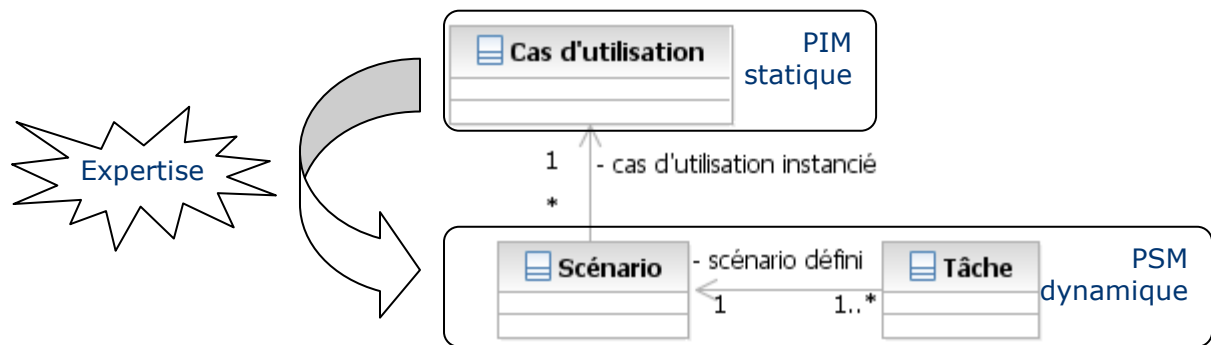


Figure 64 : Concepts et transformation liés à l'activité de collecte des exigences fonctionnelles dynamiques

Un mode de représentation des scénarios est une description textuelle avec un ordonnancement des tâches définissant une interaction entre l'utilisateur du système et le système. Pour chaque cas d'utilisation, un ensemble de scénarios détaille l'ordonnancement des tâches.

Le seul branchement conditionnel du cas d'utilisation *CU-Réception de message avec filtrage* est le résultat du filtrage du message. Il existe donc un scénario avec un filtrage ayant un résultat positif et un scénario avec un filtrage ayant un résultat négatif.

Le scénario *SC-Filtrage négatif* où le résultat du filtrage est négatif est le suivant :

1. L'utilisateur reçoit un message
2. Le résultat du filtrage du message par rapport à son objet est négatif
3. Le message est stocké dans une boîte de messagerie

Le scénario *SC-Filtrage positif* où le résultat du filtrage est positif est :

1. L'utilisateur reçoit un message
2. Le résultat du filtrage du message par rapport à son objet est positif
3. La réception du message est interrompue

La Règle d'urbanisme - Collecte des exigences fonctionnelles dynamiques est satisfaite puisque les deux scénarios sont déduits des deux résultats possibles du branchement conditionnel lié au filtrage : négatif ou positif.

4.2.1.3 Collecte des exigences non fonctionnelles

L'activité de collecte des exigences non fonctionnelles, entourée dans la **Figure 65**, a pour objectif de spécifier les exigences non fonctionnelles du système.

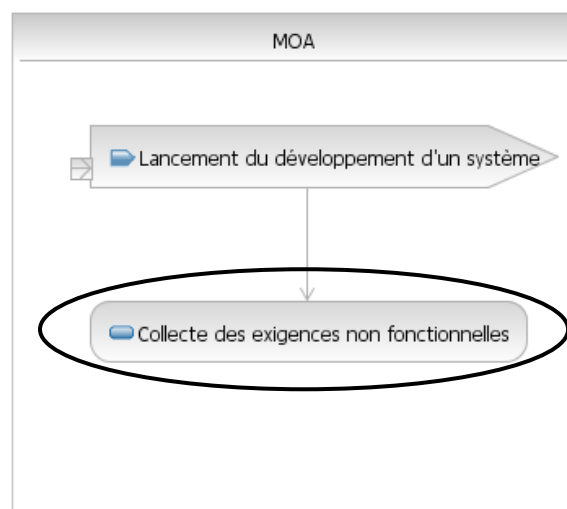


Figure 65 : Activité de collecte des exigences non fonctionnelles

Le concept de la vue technique produit par l'activité est l'exigence non fonctionnelle.

Le mode de représentation des exigences non fonctionnelles est une définition textuelle. Cette définition peut être associée à un cas d'utilisation lors de la collecte des exigences non fonctionnelles.

Les exigences non fonctionnelles dans le cadre du cas d'utilisation *CU-Réception de message avec filtrage* sont :

- *ENF-MMR* qui stipule le nombre moyen de messages reçus par utilisateur,
- *ENF-TMS* indiquant la taille maximum de stockage des messages dans une boîte de messagerie.

4.2.2 Analyse

L'analyse fonctionnelle et l'analyse technique du système à développer constituent une macro-activité sous la responsabilité de l'équipe de développement du système, soit la MOE. Cette macro-activité, entourée dans la **Figure 66**, nécessite néanmoins de nombreux échanges avec la MOA afin de s'assurer que l'analyse répond bien aux exigences collectées.

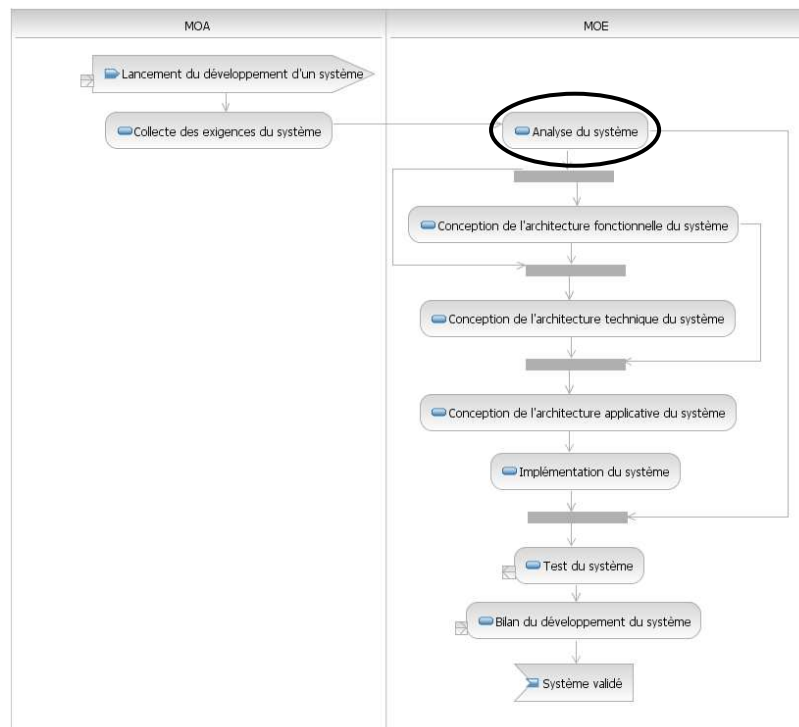


Figure 66 : Macro-activité d'analyse dans la démarche EA4UP

4.2.2.1 Analyse fonctionnelle statique

L'activité d'analyse statique entourée dans la **Figure 67** permet de spécifier les entités participantes au système et leurs relations. Les entités participantes à l'ensemble des cas d'utilisation du système, collectés lors de l'activité liées aux exigences fonctionnelles statiques (cf. §4.2.1.1), sont détaillées par attribut. Cette activité d'expert permet la définition des données de la vue fonctionnelle.

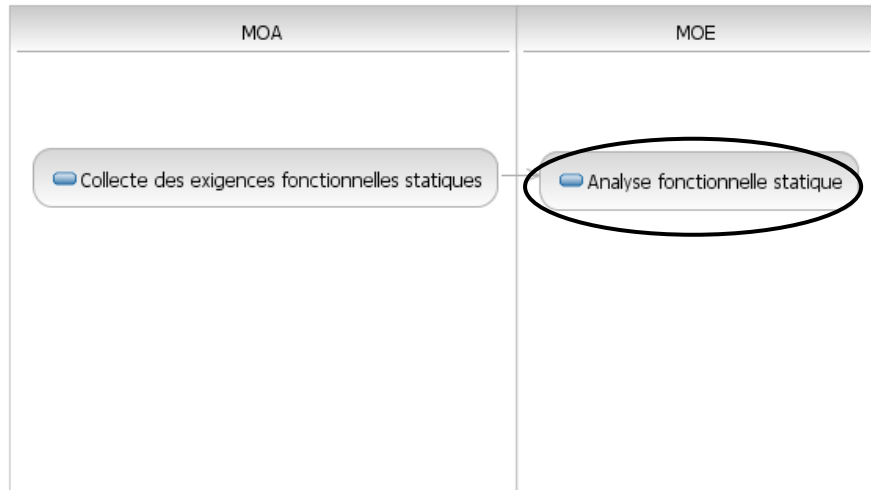


Figure 67 : Activité d'analyse fonctionnelle statique

La transformation décrite dans la **Figure 68** permet le contrôle de la cohérence de l'activité. L'activité d'analyse fonctionnelle statique est la transformation par un expert d'un ensemble de cas d'utilisation d'un système en un ensemble d'entités définies par des attributs et pouvant être reliées entre elles.

La règle d'urbanisme liée à cette activité permet de s'assurer de la traçabilité du lien entre le cas d'utilisation du système et les entités participantes à ce cas d'utilisation.

Règle d'urbanisme – Analyse fonctionnelle statique n°8

Une relation spécifiée entre deux entités nécessite la participation de ces entités à au moins un même cas d'utilisation.

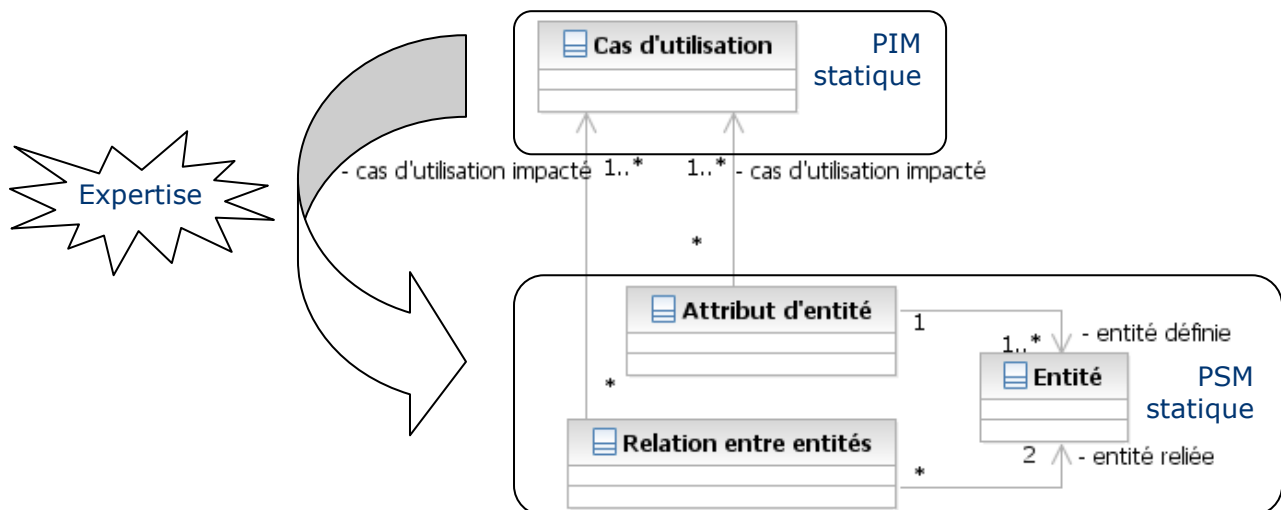


Figure 68 : Concepts et transformation liés à l'activité d'analyse fonctionnelle statique

Un mode de représentation des entités est un diagramme de classes UML stéréotypées <<entité>>. L'attribut d'une entité est un attribut de la classe représentant l'entité et une relation entre entités est une association entre les classes représentant les entités reliées.

Un exemple d'entités participantes au cas d'utilisation *CU-Réception de message avec filtrage* est représenté dans le diagramme de classes UML de la **Figure 69**.

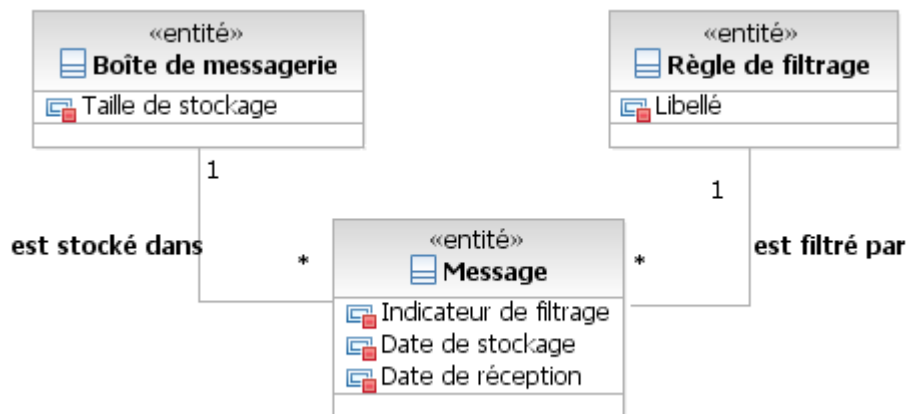


Figure 69 : Illustration des entités de l'activité d'analyse fonctionnelle statique

Lorsqu'un message est reçu, l'application de la règle de filtrage par rapport à l'objet du message permet de valoriser l'indicateur de filtrage du message. Si l'indicateur est négatif, alors le message est stocké dans une boîte de messagerie dont la taille augmente.

La Règle d'urbanisme – Analyse fonctionnelle statique est satisfaite puisque toutes les entités participent au même cas d'utilisation *CU-Réception de message avec filtrage*.

4.2.2.2 Analyse fonctionnelle dynamique

L'activité d'analyse fonctionnelle dynamique, entourée dans la **Figure 70**, permet de spécifier les interactions instanciées entre entités issues de l'analyse fonctionnelle statique (cf. §4.2.2.1). L'instanciation est liée au déroulement de scénarios collectés lors de l'activité liée aux exigences fonctionnelles dynamiques (cf. §4.2.1.2).

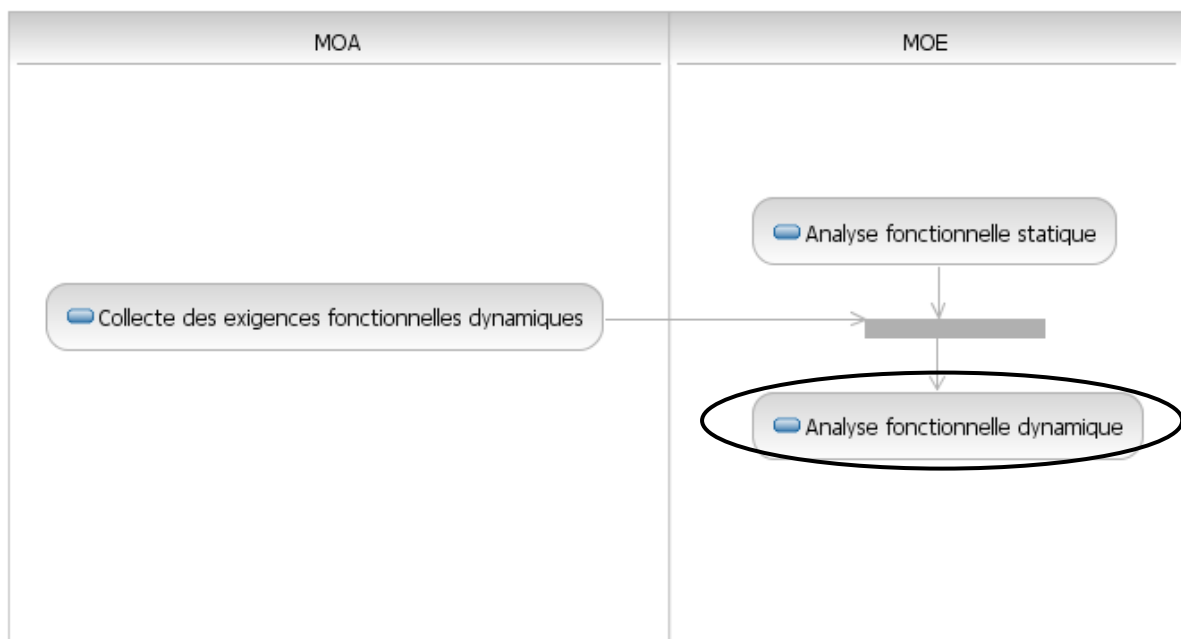


Figure 70 : Activité d'analyse fonctionnelle dynamique

Le concept de la vue fonctionnelle résultant de l'activité d'analyse fonctionnelle dynamique est l'interaction entre entités. La transformation décrite dans la **Figure 71** permet de produire les interactions entre entités à partir d'une activité de codage (cf. §4.1.3). L'expert code chaque tâche des scénarios avec les relations entre entités spécifiées en analyse fonctionnelle statique.

La règle d'urbanisme liée à cette activité permet de souligner la cohérence de l'analyse fonctionnelle dynamique avec l'analyse fonctionnelle statique.

Règle d'urbanisme – Analyse fonctionnelle dynamique n°9

Toutes les interactions entre deux entités, instanciées lors du déroulement d'un scénario illustrant l'usage d'un système, nécessitent une relation entre les deux entités.

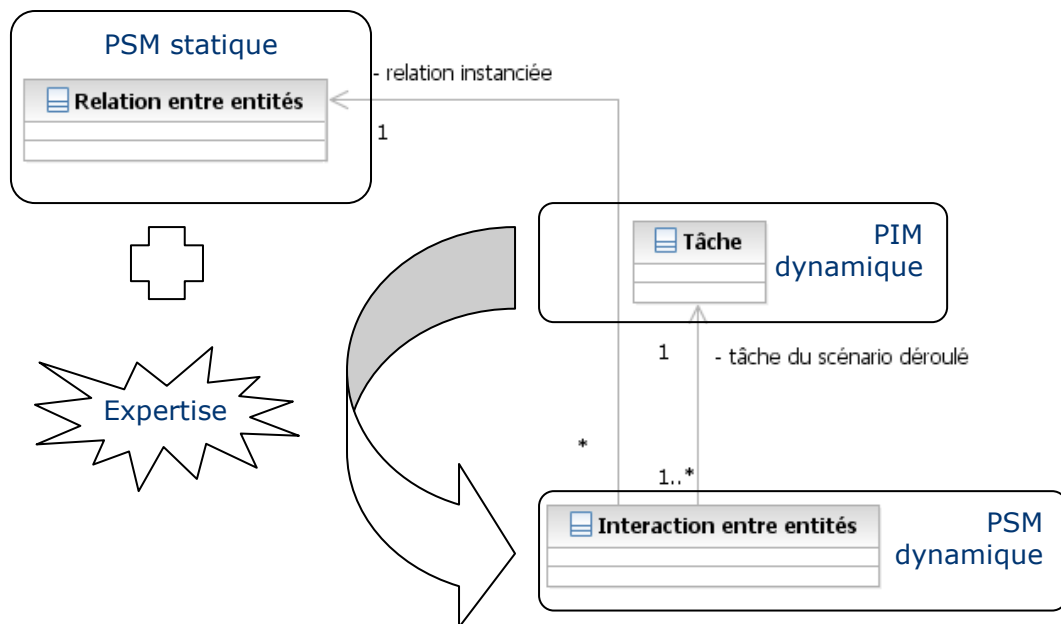


Figure 71 : Concepts et transformation liés à l'activité d'analyse fonctionnelle dynamique

Un mode de représentation des interactions entre entités instanciées lors d'un scénario de l'utilisation d'un système est le diagramme de séquence UML. La représentation de la succession temporelle d'interactions et la représentation de la dépendance d'interactions respecte la Sémantique UML – Approche dynamique et diagramme de séquence (cf. §2.1.5).

Les interactions sont de type requête ou de type réponse. La définition du type d'une interaction entre entités est analogue à celle des interactions entre îlots fonctionnels du §3.2.

Définition : une interaction entre instances d'entités est :

- de type requête si l'interaction est une demande, ou si l'interaction correspond à l'émission de paramètres d'entrée d'une requête,
- de type réponse si l'interaction est une réponse à une demande ou si l'interaction correspond à l'émission de paramètres de sortie d'une requête.

Dans l'illustration choisie, les tâches du scénario *SC-Filtrage négatif* sont représentées dans le diagramme de séquence de la **Figure 72**. Les paramètres d'entrée et de sortie d'une interaction sont indiqués entre parenthèses, respectivement, après le mot clé *in* et après le mot clé *out*. Ces paramètres sont les attributs des entitésinstanciées.

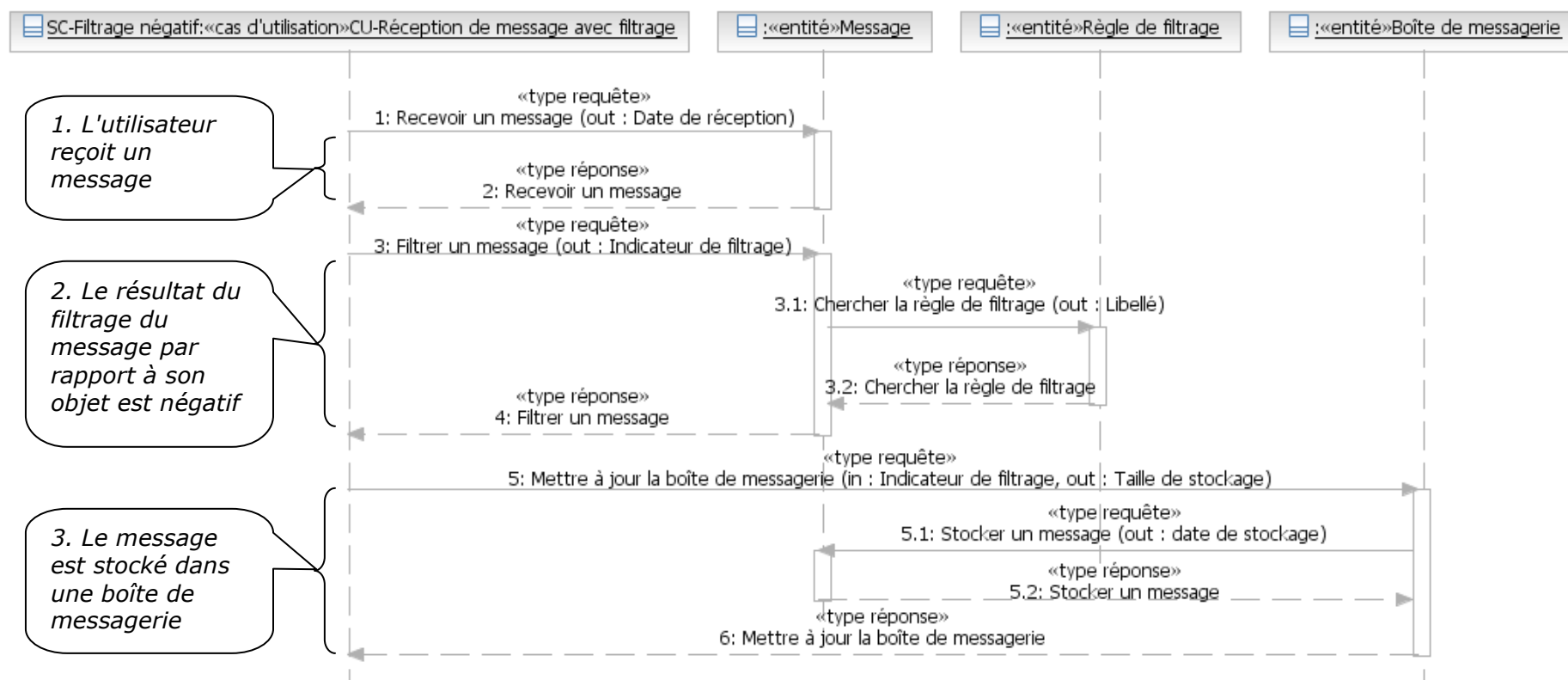


Figure 72 : Illustration des interactions de l'activité d'analyse fonctionnelle dynamique

Le lien entre chaque tâche du scénario et les entités sont les suivants :

- la tâche 1 du scénario *L'utilisateur reçoit un message* et
 - o l'entité *Message* et son attribut *Date de réception*,
- la tâche 2 du scénario *Le résultat du filtrage du message par rapport à son objet est négatif* et
 - o l'entité *Message* et son attribut *Indicateur de filtrage*
 - o l'entité *Règle de filtrage* et son attribut *Libellé*,
- la tâche 3 du scénario *Le message est stocké dans une boîte de messagerie* et
 - o l'entité *Message* et son attribut *Date de stockage*
 - o l'entité *Boîte de messagerie* et son attribut *Taille de stockage*.

La Règle d'urbanisme – Analyse fonctionnelle dynamique est satisfaite puisque :

- les interactions entre les instances des entités *Boîte de messagerie* et *Message* sont conformes à la relation *est stocké dans* entre ces entités,
- les interactions entre les instances des entités *Message* et *Règle de filtrage* sont conformes à la relation *est filtré par* entre ces entités.

4.2.2.3 Analyse technique statique

L'activité d'analyse technique statique, entourée dans la **Figure 73**, permet d'associer une exigence non fonctionnelle collectée (cf. §4.2.1.3) à un élément spécifié lors de l'analyse fonctionnelle statique (cf. §4.2.2.1).

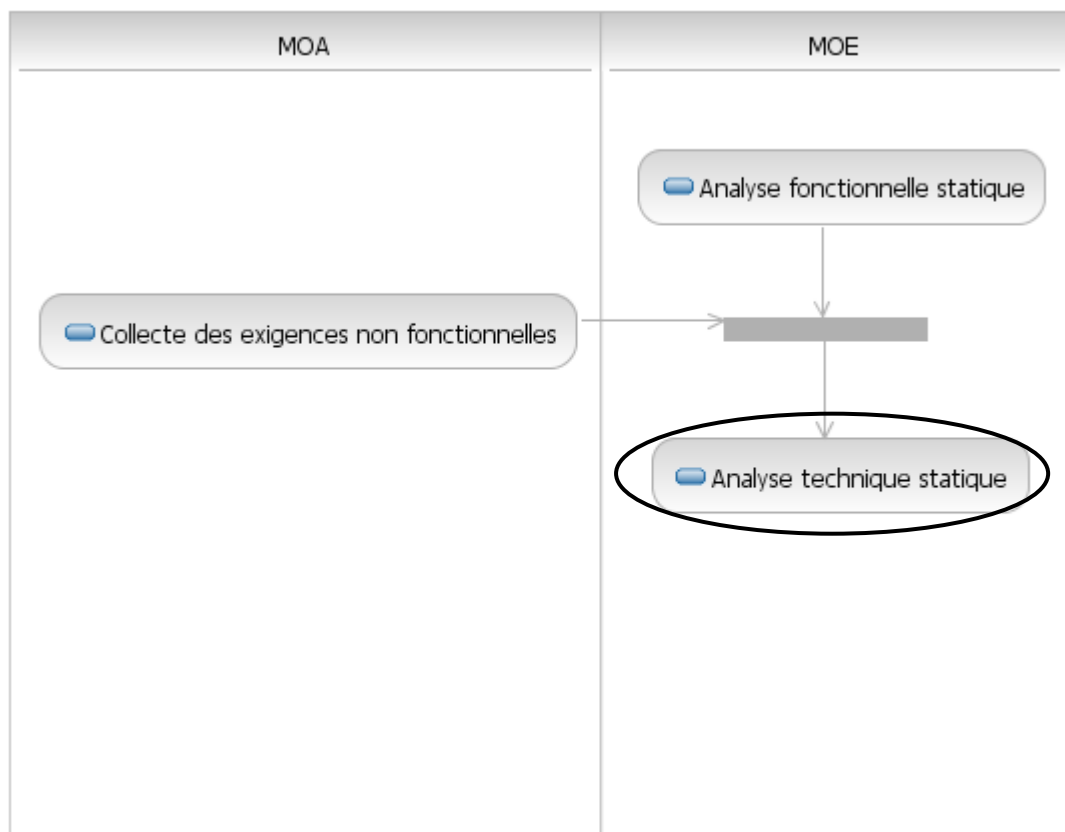


Figure 73 : Activité d'analyse technique statique

Le concept défini par l'association d'une exigence non fonctionnelle avec une entité ou une relation entre entités est une propriété technique statique. Ce concept de la vue technique décrit dans la **Figure 74** permet de préciser les éléments spécifiés durant l'analyse fonctionnelle statique qui sont impactés par une exigence non fonctionnelle.

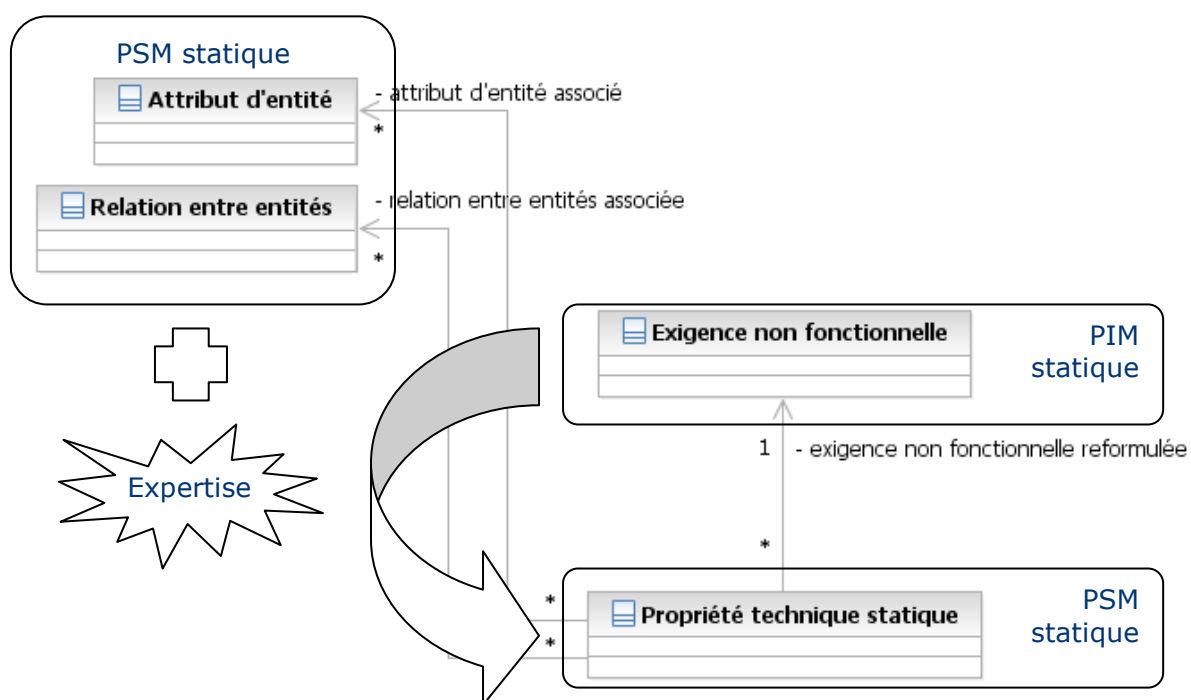


Figure 74 : Concepts et transformation liés à l'activité d'analyse technique statique

Un mode de représentation du concept de propriété technique statique est une description textuelle.

Le Tableau 12 représente les exigences non fonctionnelles associées aux attributs d'entité ou aux relations entre entités pour le cas d'utilisation *CU-Réception de message avec filtrage*. Chaque propriété technique statique est un couple formé par un élément d'analyse fonctionnelle statique et une exigence non fonctionnelle collectée.

Élément d'analyse fonctionnelle statique			Exigence non fonctionnelle
Entité	Attribut d'entité	Lien entre entités	
Message	Date de réception		ENF-MMR
	Indicateur de filtrage		ENF-MMR
Boîte de messagerie	Taille de stockage		ENF-TMS
Message		est stocké dans	ENF-TMS
Boîte de messagerie			

Tableau 12 : Illustration des propriétés techniques statiques de l'activité d'analyse technique statique.

Un exemple de propriété technique statique est le couple (*Date de réception*, *ENF-MMR*). L'exigence *ENF-MMR* du nombre moyen de messages reçus impacte en effet l'attribut *Date de réception* de l'entité *Message*. Un autre exemple est le couple (*est stocké dans*, *ENF-TMS*) qui décrit le lien entre

- l'exigence *ENF-TMS* indiquant la taille maximum de la boîte de messagerie, et
- la relation *est stocké dans* qui spécifie le stockage d'un message dans cette boîte.

4.2.2.4 Analyse technique dynamique

L'activité d'analyse technique dynamique, entourée dans la **Figure 75**, permet d'associer des exigences non fonctionnelles avec des interactions de type requête instanciées lors du déroulement de scénarios de cas d'utilisation du système (cf. §4.2.2.2). L'association est déduite de l'analyse technique statique (cf. §4.2.2.3) où les attributs des entités et les relations entre entités sont associés à des exigences non fonctionnelles. Une interaction de type requête entre instances d'entités doit en effet être conforme à une relation entre entités d'après la Règle d'urbanisme – Analyse fonctionnelle dynamique. De plus, le paramètre de sortie d'une interaction de type requête est l'instanciation d'au moins un attribut d'une entité.

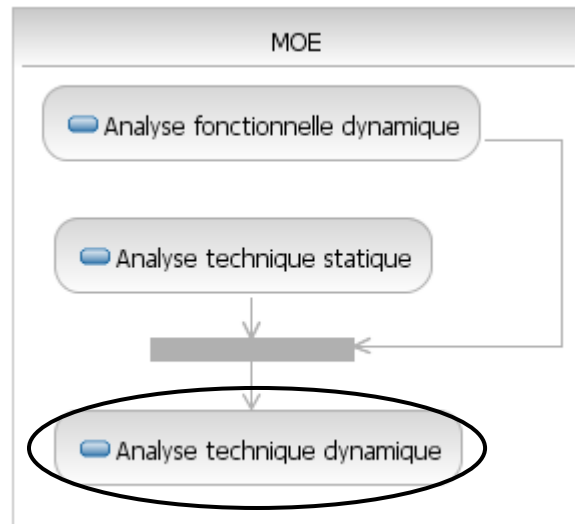


Figure 75 : Activité d'analyse technique dynamique

Le concept de la vue technique résultant de l'activité est la propriété technique dynamique. La transformation décrite dans la **Figure 76** permet de produire les propriétés techniques dynamiques. Cette transformation a l'ensemble des propriétés statiques en entrée et est contrainte par les interactions entre entités définies lors de l'analyse fonctionnelle dynamique.

La règle d'urbanisme liée à cette activité permet de souligner la cohérence de l'analyse technique statique avec l'analyse technique dynamique.

Règle d'urbanisme – Analyse technique dynamique n°10

Toutes propriétés techniques dynamiques liée à une interaction entre entités de type requête est associée :

- à une propriété technique statique liée à une relation entre les entités de l'interaction, ou
- à une propriété technique statique liée à un attribut de l'entité cible de l'interaction.

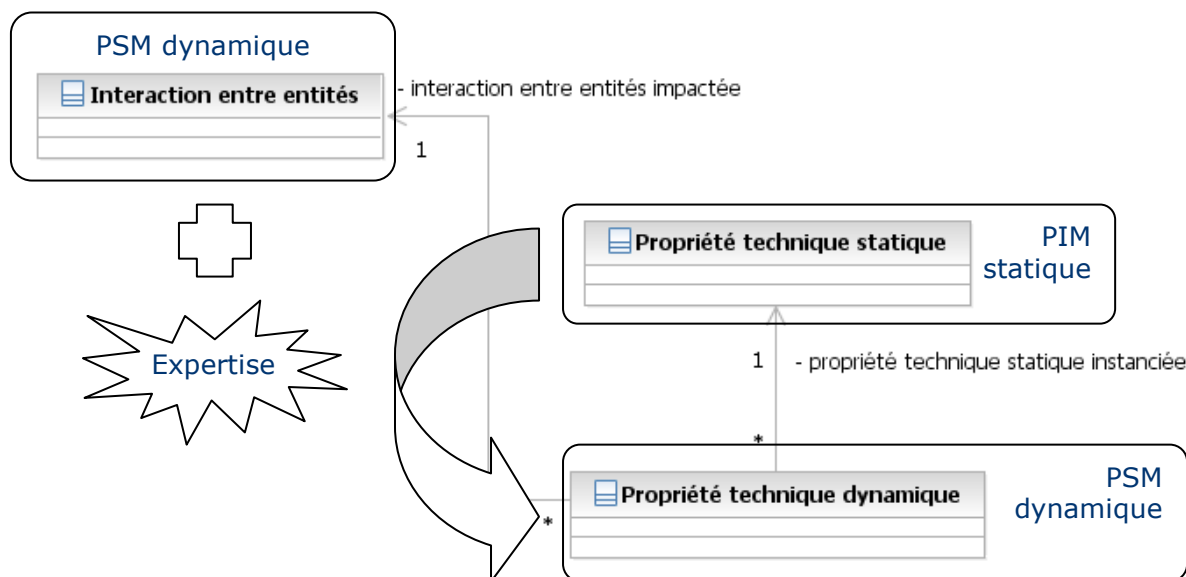


Figure 76 : Concepts et transformation liés à l'activité d'analyse technique dynamique

Un mode de représentation du concept de propriété technique dynamique est une description textuelle ajoutée au diagramme de séquence UML qui représente les interactions entre instances d'entités.

Le diagramme de séquence de la **Figure 77** représente le scénario *SC-Filtrage négatif* auquel sont intégrées l'exigence non fonctionnelle *ENF-MMR* du nombre moyen de messages reçus et l'exigence non fonctionnelle *ENF-TMS* de la taille maximale de stockage de messages.

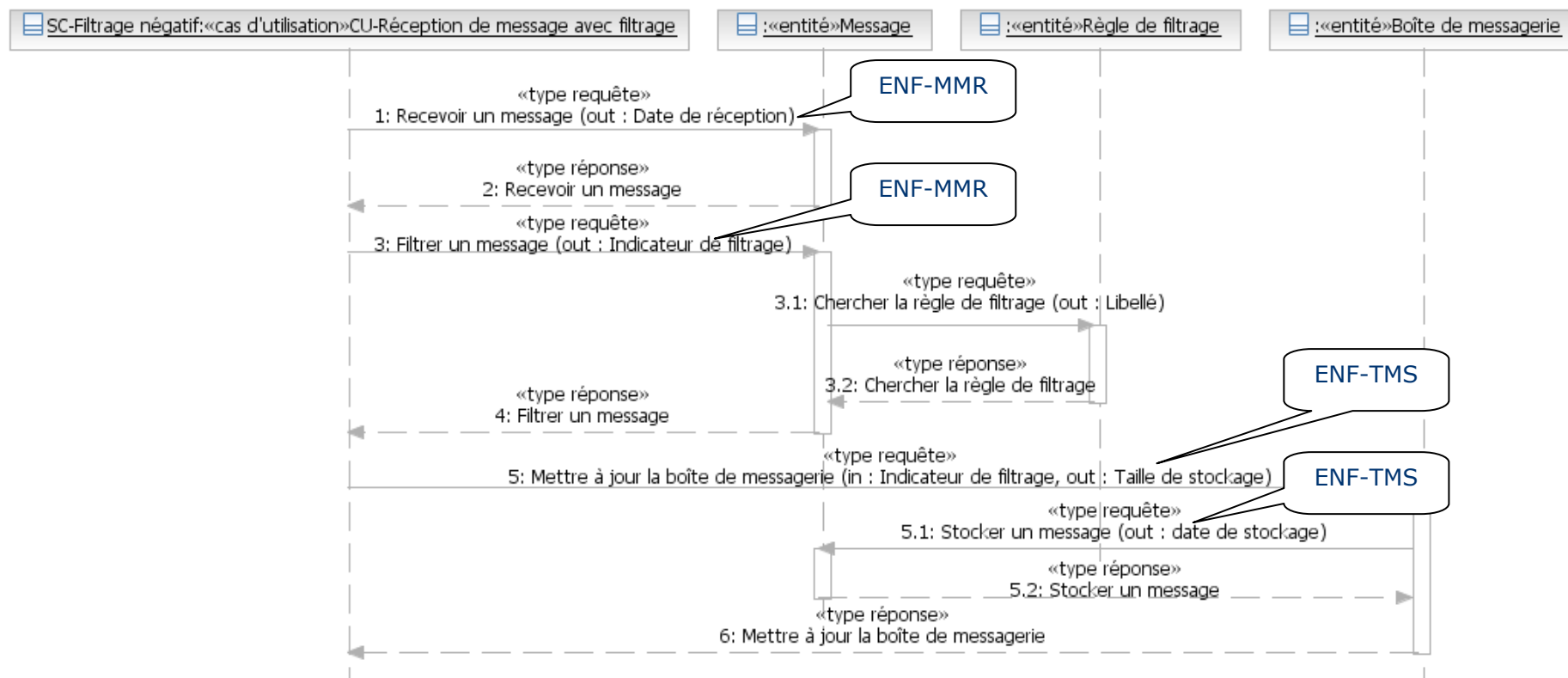


Figure 77 : Illustration des propriétés techniques dynamiques de l'activité d'analyse technique dynamique

Un exemple de propriété technique dynamique est l'association entre la propriété technique statique (*Date de réception, ENF-MMR*) et l'interaction de type requête *Recevoir un message*. Une autre propriété technique dynamique est l'association entre la propriété technique statique (*est stocké dans, ENF-TMS*) et l'interaction de type requête *Stocker un message*.

La Règle d'urbanisme – Analyse technique dynamique (cf. §4.2.4.2) est satisfaite puisque toutes les propriétés techniques dynamiques sont associées directement à des propriétés techniques statiques.

4.2.3 Conception de l'architecture fonctionnelle

La conception de l'architecture fonctionnelle du système est une macro-activité sous la responsabilité de l'équipe de développement, soit la MOE du système. Cette macro-activité, entourée dans la **Figure 78**, nécessite que le système soit analysé. Le PLU fonctionnel conçu par les urbanistes fonctionnels contraint les activités d'architecture fonctionnelle.

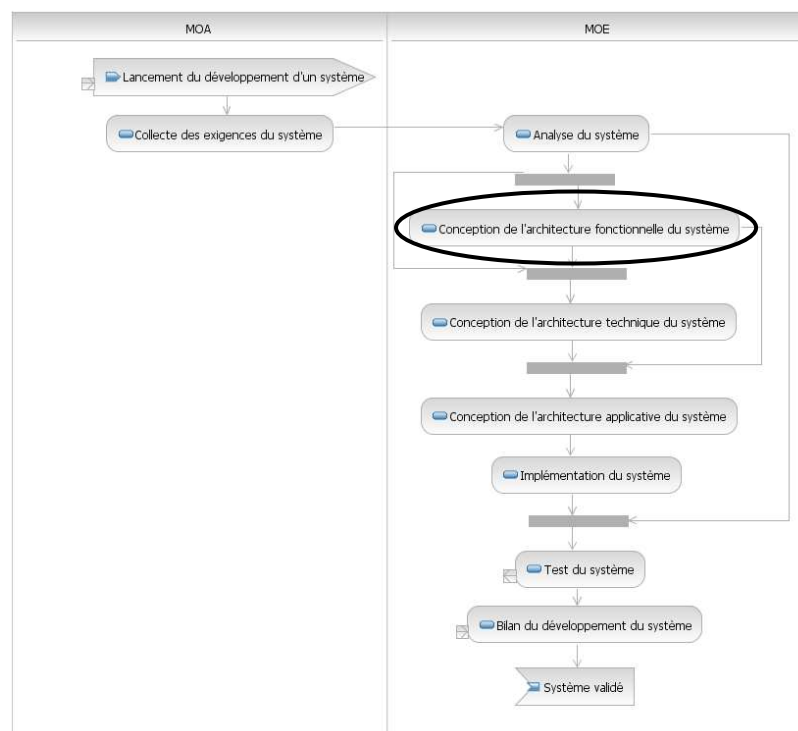


Figure 78 : Macro-activité de conception de l'architecture fonctionnelle dans la démarche EA4UP

4.2.3.1 Architecture fonctionnelle statique

L'activité de conception de l'architecture fonctionnelle statique d'un système, entourée dans la **Figure 79**, a pour objectif de concevoir les îlots de la vue fonctionnelle du système qui réalisent les éléments spécifiés lors de l'analyse fonctionnelle statique (cf. §4.2.2.1). Les îlots sont soit extraits du PLU fonctionnel validé (cf. 3.1.2), soit créés si nécessaire. La vue fonctionnelle d'éléments à réutiliser du SI doit aussi être comparée avec la vue fonctionnelle du système lors de cette activité.

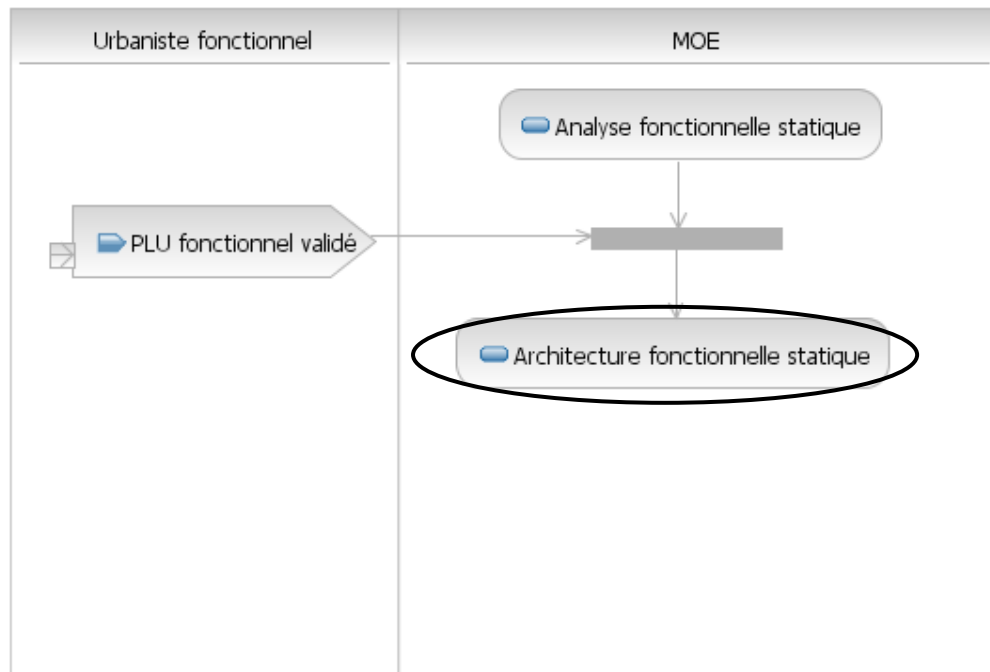


Figure 79 : Activité d'architecture fonctionnelle statique

Le concept principal de l'activité d'architecture fonctionnelle statique est la parcelle fonctionnelle représentée dans la **Figure 80**. L'expert code chaque attribut d'entité, ou chaque relation entre entités, avec une ou plusieurs parcelles fonctionnelles. Ce codage signifie qu'une donnée produite par cette ou ces parcelles fonctionnelles est extraite de l'attribut ou de la relation codée. Chacune de ces parcelles est contenue dans un îlot fonctionnel. L'îlot fonctionnel est soit issu du PLU fonctionnel afin de favoriser la réutilisation de tout ou partie du système (cf. §4.1.1), soit spécifique au système.

La règle d'urbanisme associée à l'activité d'architecture fonctionnelle statique assure la cohésion avec l'architecture du PLU fonctionnel. La cohésion est due à l'adéquation de chaque parcelle avec le typage de l'îlot qui la contient.

Règle d'urbanisme – Architecture fonctionnelle statique n°11

Chaque parcelle fonctionnelle codant un attribut d'une entité ou une relation entre entités doit être contenue :

- dans un îlot fonctionnel de type flux, suivant la dépendance de la durée de vie d'une instance de la parcelle par rapport au processus supporté (cf. §3.3.1),
- dans un îlot fonctionnel de type stock suivant l'indépendance de la durée de vie d'une instance de la parcelle par rapport au processus supporté (cf. §3.3.1).

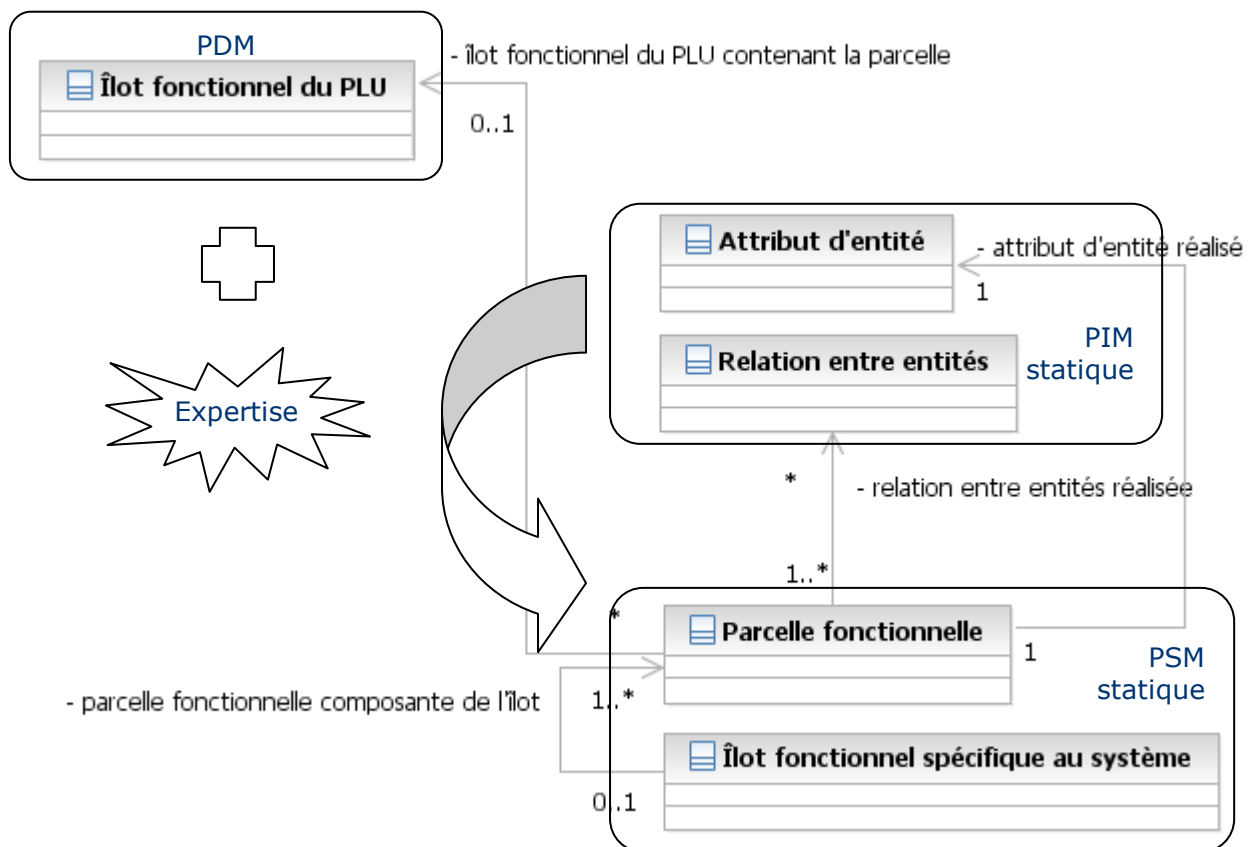


Figure 80 : Concepts et transformation liés à l'activité d'architecture fonctionnelle statique

La contrainte de l'association d'une parcelle fonctionnelle et d'un îlot fonctionnel qui le contient est que chaque parcelle doit appartenir de façon exclusive :

- soit à un îlot fonctionnel unique conçu dans le PLU fonctionnel,
- soit à un îlot spécifique au système développé.

Le mode de représentation des îlots fonctionnels et de leurs parcelles fonctionnelles est le diagramme de classes UML décrit dans le §3.2 avec le stéréotype défini dans le §3.3.2.

L'expert code les éléments de l'analyse fonctionnelle statique avec les parcelles du PLU fonctionnel représenté dans la **Figure 31** ou avec des parcelles fonctionnelles spécifiques au service. Le Tableau 13 comprend les résultats de ce codage.

Élément d'analyse fonctionnelle statique			Parcelle fonctionnelle du PLU	Parcelle fonctionnelle spécifique
Entité	Attribut d'entité	Lien entre entités		
Message	Date de réception		Recevoir un message	
	Indicateur de filtrage		Filtrer un message par rapport à l'objet du message	
	Date de stockage			Stocker un message
Boîte de messagerie	Taille de stockage			
Règle de filtrage	Libellé			Consulter une règle de filtrage de message
Message		est stocké dans		Stocker un message
Boîte de messagerie				
Message		est filtré par	Filtrer un message par rapport à l'objet du message	
Règle de filtrage				

Tableau 13 : Illustration du codage des éléments d'analyse fonctionnelle statique par les parcelles fonctionnelles du système.

Le choix de conception de l'architecte fonctionnel est ici de ne pas créer de nouvel îlot fonctionnel contenant les parcelles fonctionnelles non décrites dans le PLU fonctionnel. Les parcelles fonctionnelles spécifiques sont regroupées avec des parcelles fonctionnelles du PLU fonctionnellement proches.

La parcelle *Stocker un message* est, par exemple, proche fonctionnellement de la parcelle *Recevoir un message*. La parcelle fonctionnelle *Stocker un message* est donc incluse par l'expert dans l'îlot fonctionnel *Réception d'un message*. Il en est de même de la parcelle fonctionnelle du PLU fonctionnel *Créer les règles de filtrage de message* et de la parcelle fonctionnelle spécifique du service *Consulter une règle de filtrage de message* qui sont assez proches fonctionnellement pour être regroupées dans l'îlot fonctionnel *Sécurité de la messagerie – Gérer les règles de filtrage*.

Les classes UML de la **Figure 81** représente les îlots fonctionnels et les parcelles fonctionnelles réalisant les éléments d'analyse fonctionnelle statique qui participent au cas d'utilisation du système *CU-Réception de message avec filtrage*.

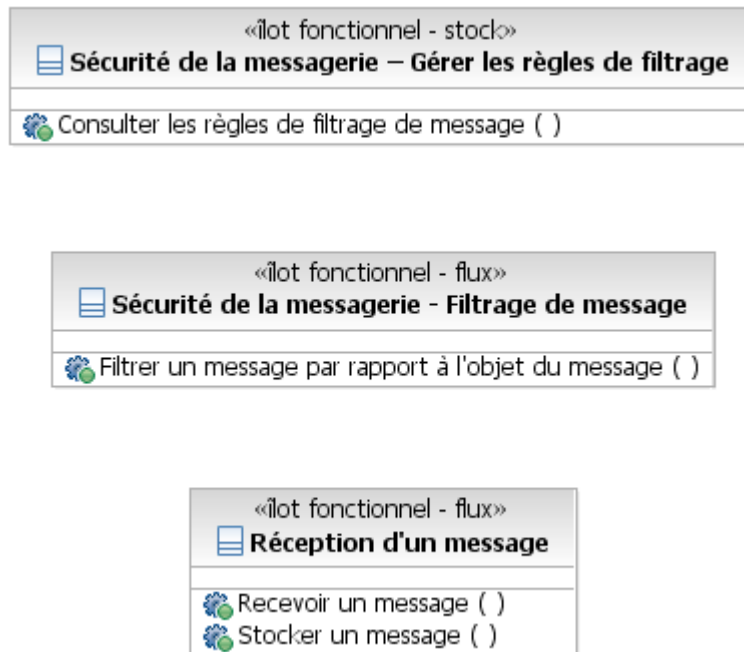


Figure 81 : Illustration des îlots fonctionnels de l'activité d'architecture fonctionnelle statique

La Règle d'urbanisme – Architecture fonctionnelle statique est respectée dans cette illustration puisque

- la parcelle fonctionnelle *Stocker un message* doit appartenir à un îlot fonctionnel de type flux et,
- la parcelle fonctionnelle *Consulter une règle de filtrage de message* doit appartenir à un îlot fonctionnel de type stock.

De plus, les parcelles fonctionnelles conçues correspondent à la vue fonctionnelle de deux enablers :

- l'enabler *Stockage* pour la parcelle fonctionnelle *Stocker un message*,
- l'enabler *Messagerie* pour toutes les autres parcelles.

4.2.3.2 Architecture fonctionnelle dynamique des îlots

L'activité de conception de l'architecture fonctionnelle dynamique des îlots d'un système conçus durant l'activité d'architecture fonctionnelle statique (cf. §4.2.3.1), entourée dans la **Figure 82**, a pour objectif de concevoir les interactions entre îlots de la vue fonctionnelle du système. Ces interactions entre îlots réalisent un scénario spécifié en analyse fonctionnelle dynamique (cf. §4.2.2.2). Les interactions entre îlots sont soit conformes au PLU fonctionnel validé (cf. 3.1.2), soit créées lorsqu'elles ne sont pas décrites dans le PLU fonctionnel.

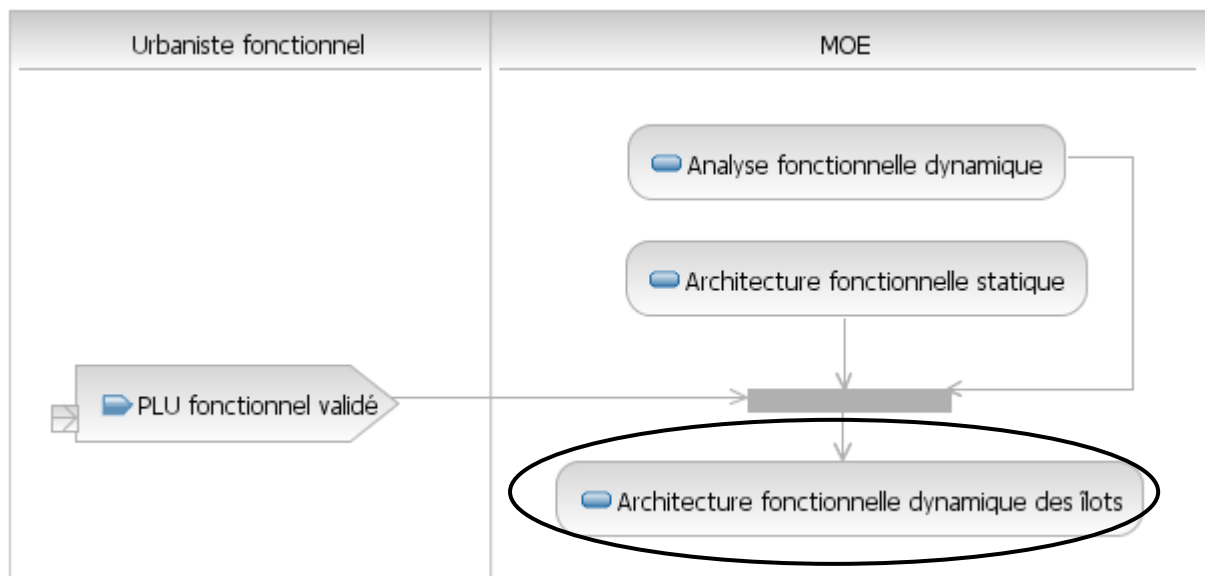


Figure 82 : Activité d'architecture fonctionnelle dynamique des îlots

Le concept principal de l'activité d'architecture fonctionnelle dynamique des îlots décrit dans la **Figure 85** est l'interaction entre îlots fonctionnels. Cette interaction est instanciée lors du déroulement d'un scénario d'un cas d'utilisation du système. Le second concept déduit immédiatement des interactions entre îlots fonctionnels est la voie fonctionnelle entre ces îlots.

L'architecture fonctionnelle dynamique des îlots est une transformation automatisée conforme à l'enchaînement décrit dans la **Figure 59**. Le codage est celui des attributs des entités et des relations entre entités par les parcelles fonctionnelles lors de l'activité d'architecture fonctionnelle statique (cf. §4.2.3.1). En complément des parcelles fonctionnelles conçues lors de l'architecture fonctionnelle statique, l'existence et l'orientation des voies du PLU fonctionnel contraignent la transformation.

La règle d'urbanisme associée à l'activité d'architecture fonctionnelle dynamique des îlots assure

- l'alignement des interactions entre instances de voies fonctionnelles avec une interaction entre instances d'entités,
- la cohérence des interactions entre îlots fonctionnels avec les voies fonctionnelles conçues dans le PLU fonctionnel.

Définition : Une interaction entre deux instances d'îlots fonctionnels est alignée avec une interaction entre instances d'entités si :

- les deux îlots fonctionnels codent une même entité qui est la source ou la cible d'une interaction,
- il existe une interaction entre une instance d'une entité codée par un des deux îlots fonctionnels de l'interaction et une instance d'une entité codée par l'autre îlot fonctionnel.

Un exemple simple d'alignement d'interactions entre îlots fonctionnel avec des interactions entre entités est décrit dans la **Figure 83** pour le codage et dans la **Figure 84** pour l'alignement des interactions. Les diagrammes de séquence instanciant des entités et des îlots fonctionnels représentent le même scénario du cas d'utilisation *CU*.

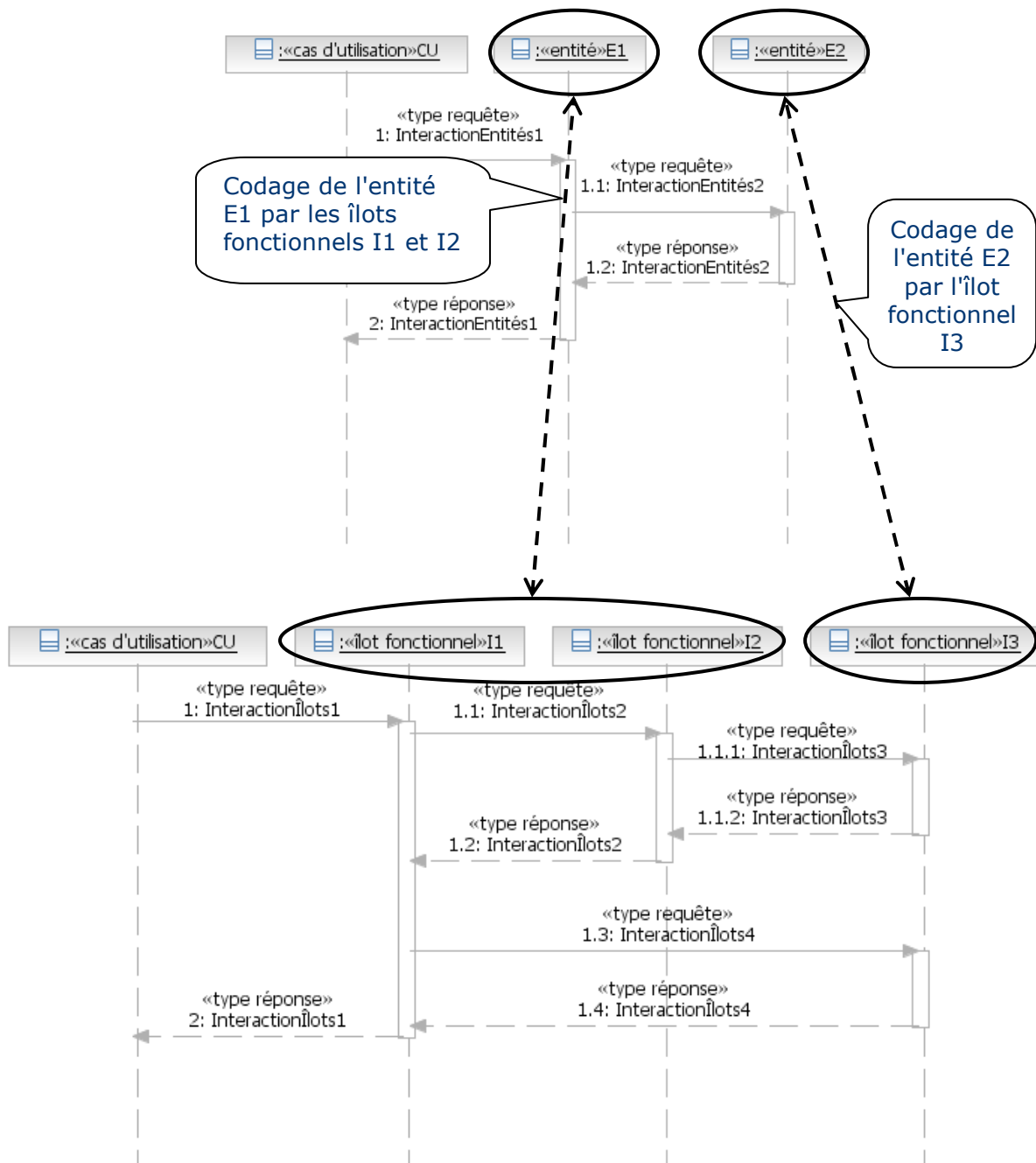


Figure 83 : Alignement d'interactions dans la règle d'urbanisme liée à l'architecture fonctionnelle dynamique des îlots – Codage

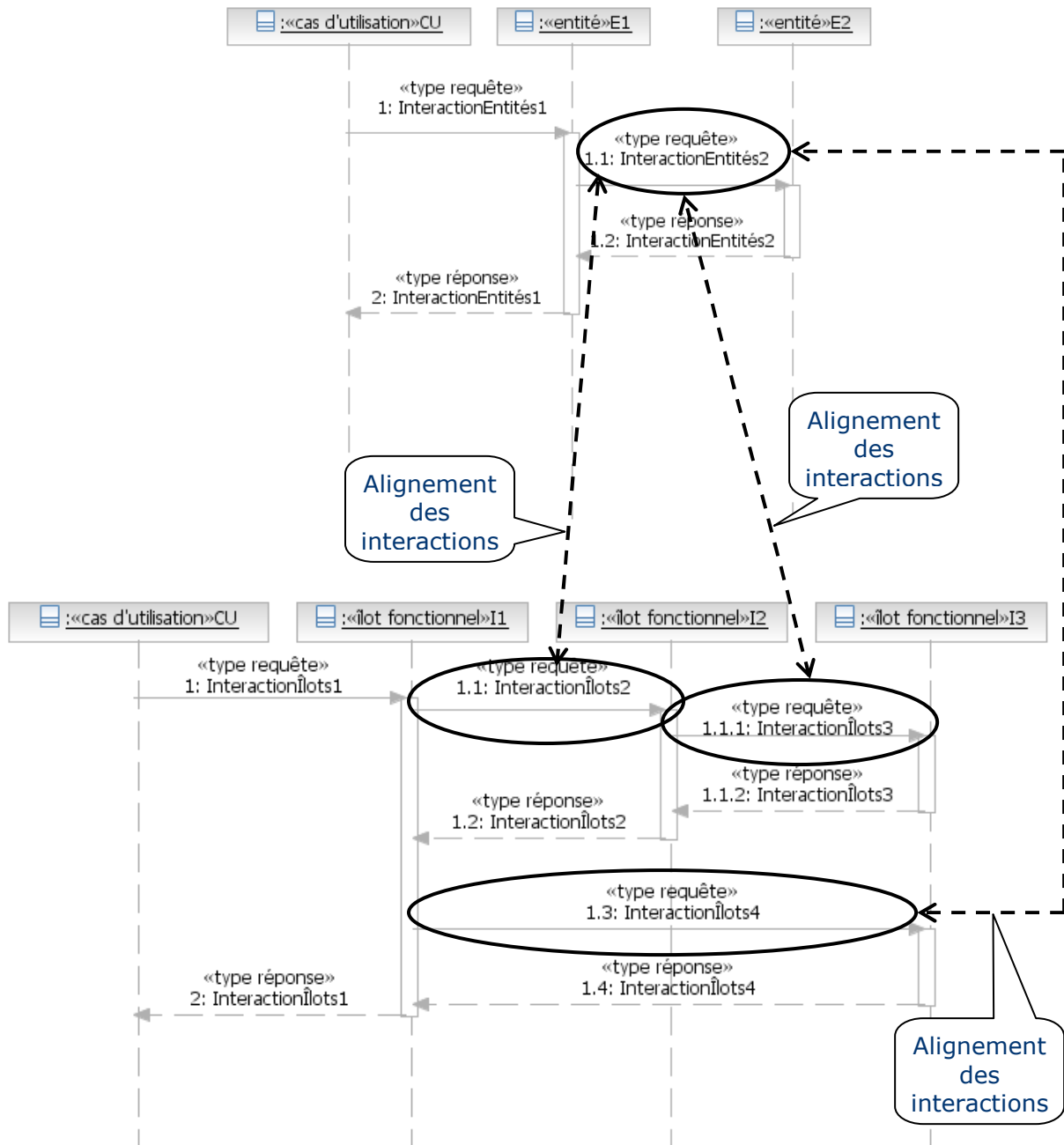


Figure 84 : Alignement d'interactions dans la règle d'urbanisme liée à l'architecture fonctionnelle dynamique des îlots – Alignement des interactions

L'interaction *InteractionÎlots2* entre les instances des îlots fonctionnels *I1* et *I2* qui codent la même entité *E1* est alignée par définition avec l'interaction *InteractionEntités2* entre les instances des entités *E1* et *E2* puisque *I1* et *I2* codent *E1*.

Toujours par définition,

- l'interaction *InteractionÎlots3* entre les instances des îlots fonctionnels *I2*, qui code l'entité *E1*, et *I3*, qui code l'entité *E2* est alignée avec l'interaction *InteractionEntités2*,
- l'interaction *InteractionÎlots4* entre les instances des îlots fonctionnels *I1*, qui code l'entité *E1*, et *I3*, qui code l'entité *E2* est alignée avec l'interaction *InteractionEntités2*.

Règle d'urbanisme – Architecture fonctionnelle dynamique des îlots n°12

Chaque interaction entre instances d'îlots fonctionnels doit :

- respecter la Règle d'urbanisme – Interactions et voies (cf. §3.2) pour les voies entre îlots extraits du PLU fonctionnel,
- intégrer la Règle d'urbanisme – Flux et stock (cf. §3.3.1) lorsqu'elle est de type requête,
- ne pas créer de cycle dans le modèle des voies fonctionnelles lorsqu'elle est de type requête,
- être alignée avec une interaction entre instances d'entités.

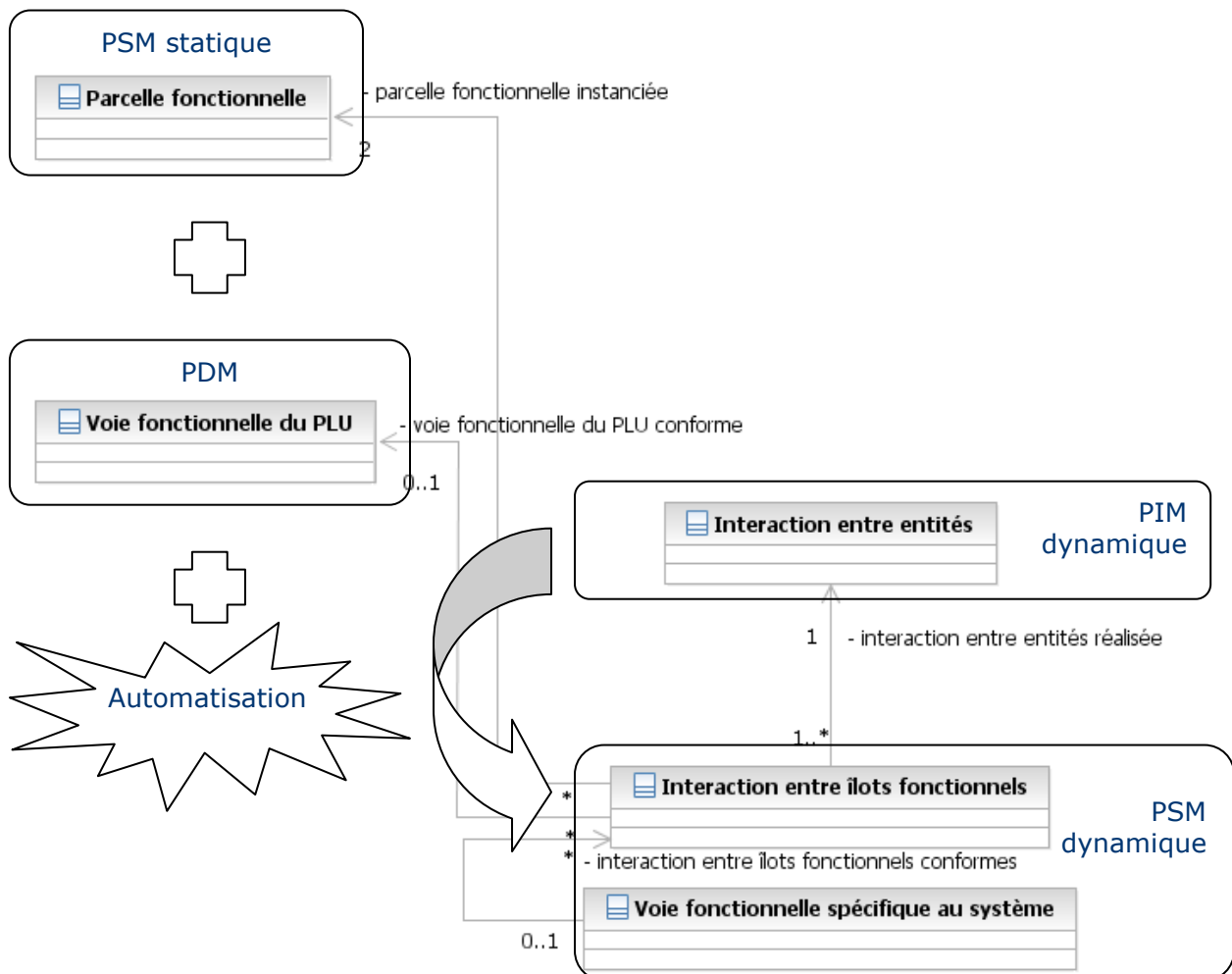


Figure 85 : Concepts et transformation liés à l'activité d'architecture fonctionnelle dynamique des îlots

D'après la Règle d'urbanisme – Interactions et voies (cf. §3.2), il est immédiat de déduire les voies entre les îlots fonctionnels, qu'elles soient extraites du PLU ou non, à partir des interactions conçues entre îlots fonctionnels.

Le mode de représentation des interactions entre instances d'îlots fonctionnels est le diagramme de séquence UML. Chaque interaction est libellée par le nom de la parcelle fonctionnelle instanciée. La représentation de la succession temporelle d'interactions et de la dépendance d'interactions respecte la Sémantique UML – Approche dynamique et diagramme de séquence (cf. §2.1.5).

Le mode de représentation des voies fonctionnelles sont des dépendances stéréotypées <<dépend de>> dans le diagramme de classes UML des îlots fonctionnels décrit dans le §3.2.

Dans la **Figure 86**, le diagramme de séquence composé d'interactions entre instances d'îlots fonctionnels représente le scénario *SC-Filtrage négatif*. Ce diagramme est le résultat de la transformation du diagramme de séquence composé d'interactions entre entités (cf. **Figure 72**). Le codage des éléments d'analyse fonctionnelle statique représenté dans le Tableau 13 permet l'automatisation de cette transformation.

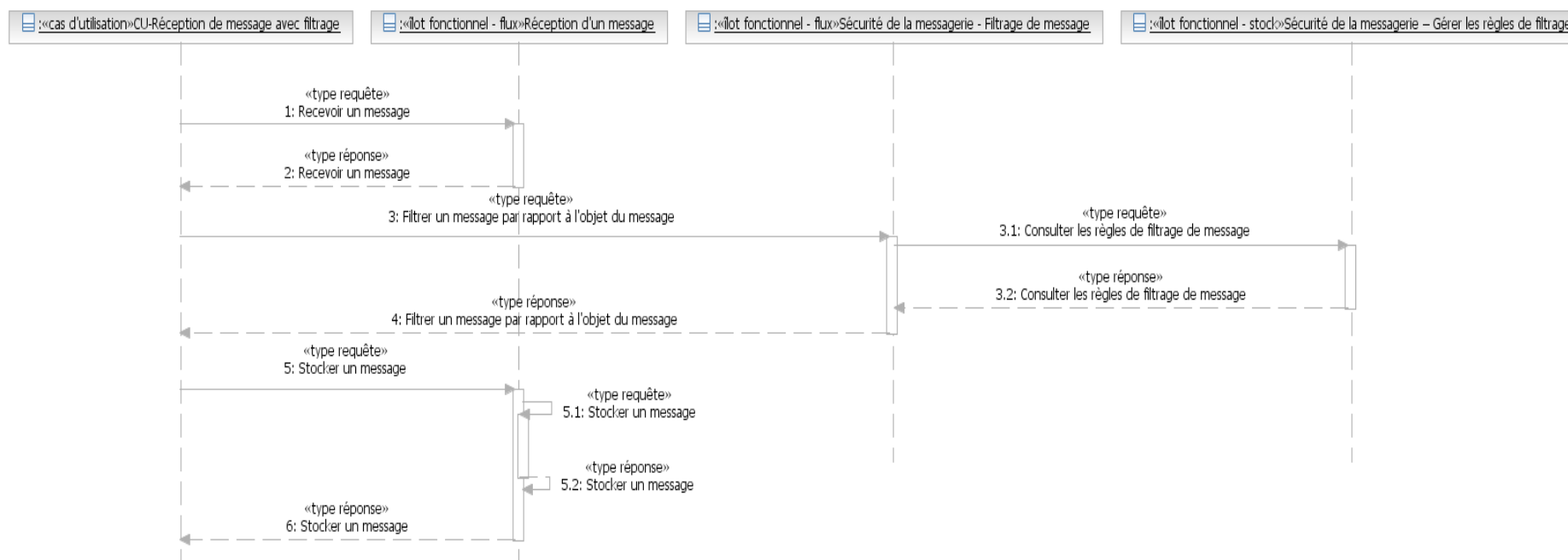


Figure 86 : Illustration des interactions entre îlots fonctionnels de l'activité d'architecture fonctionnelle dynamique des îlots

La Règle d'urbanisme – Architecture fonctionnelle dynamique des îlots est respectée puisque chaque interaction entre instances d'îlots fonctionnels est alignée avec une ou plusieurs interactions entre instances d'entités spécifiée en analyse fonctionnelle dynamique :

- l'interaction entre instances d'îlots fonctionnels *Recevoir un message* est alignée avec l'interaction entre instances d'entités du même nom,
- l'interaction entre îlots fonctionnels *Filtrer un message par rapport à l'objet du message* est alignée avec l'interaction entre instances d'entités *Filtrer un message*,
- l'interaction entre instances d'îlots fonctionnels *Consulter les règles de filtrage de message* est alignée avec l'interaction entre instances d'entités *Chercher la règle de filtrage*,
- l'interaction entre instances d'îlots fonctionnels *Stocker un message* est alignée avec l'interaction entre instances d'entités *Mettre à jour la boîte de messagerie* et avec l'interaction entre instances d'entités *Stocker un message*.

De plus les Règle d'urbanisme – Interactions et voies (cf. §3.2) pour les voies entre îlots extraits du PLU fonctionnel et la Règle d'urbanisme – Flux et stock (cf. §3.3.1) pour les interactions de type requête sont respectées :

- l'interaction de type requête entre instances d'îlots fonctionnels *Consulter les règles de filtrage de message* est orientée de l'îlot fonctionnel de type flux *Sécurité de la messagerie – Filtrage de message* vers l'îlot fonctionnel de type stock *Sécurité de la messagerie – Gérer les règles de filtrage*.

Les voies fonctionnelles cohérentes avec ces interactions entre instances d'îlots fonctionnels ne constituent aucun cycle. Ces voies satisfont de plus la Règle d'urbanisme – Interactions et voies (cf. §3.2). Le modèle d'îlots fonctionnels est représenté dans la **Figure 87**.

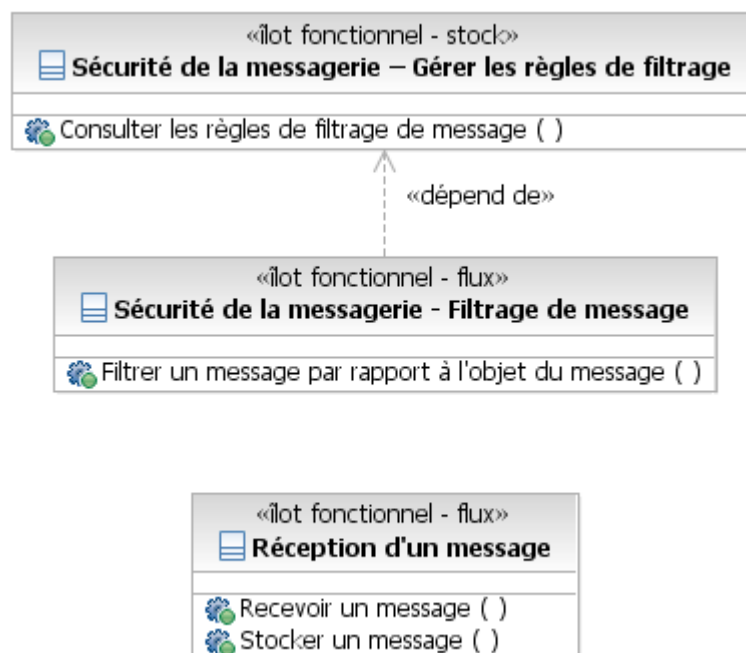


Figure 87 : Illustration des voies fonctionnelles de l'activité d'architecture fonctionnelle dynamique des îlots

Pour ce scénario *SC-Filtrage négatif*, aucun autre îlot fonctionnel que ceux décrits dans le PLU fonctionnel n'est nécessaire et seule la voie de l'îlot fonctionnel *Sécurité de la messagerie – Filtrage de message* vers l'îlot fonctionnel *Sécurité de la messagerie – Gérer les règles de filtrage* est utile.

4.2.3.3 *Architecture fonctionnelle dynamique des données*

L'activité de conception de l'architecture fonctionnelle dynamique des données d'un système, entourée dans la **Figure 88**, a pour premier objectif de concevoir les attributs des données logiques. Ceux-ci sont les paramètres de sortie des interactions entre instances d'îlots fonctionnels décrites lors de l'activité d'architecture dynamique des îlots (cf. §4.2.3.2).

Les attributs des données logiques sont :

- extraits des attributs des entités décrits en analyse fonctionnelle dynamique (cf. §4.2.2.2),
- les paramètres de sortie des interactions, de type requête, entre instances d'entités.

Le second objectif est de définir les données logiques à partir des attributs. Le respect de la Règle d'urbanisme – Relation de dépendance entre données (cf. §4.1.2) signifie que les voies fonctionnelles conçues lors de l'activité d'architecture dynamique des îlots peuvent évoluer lors de l'activité d'architecture fonctionnelle dynamique des données.

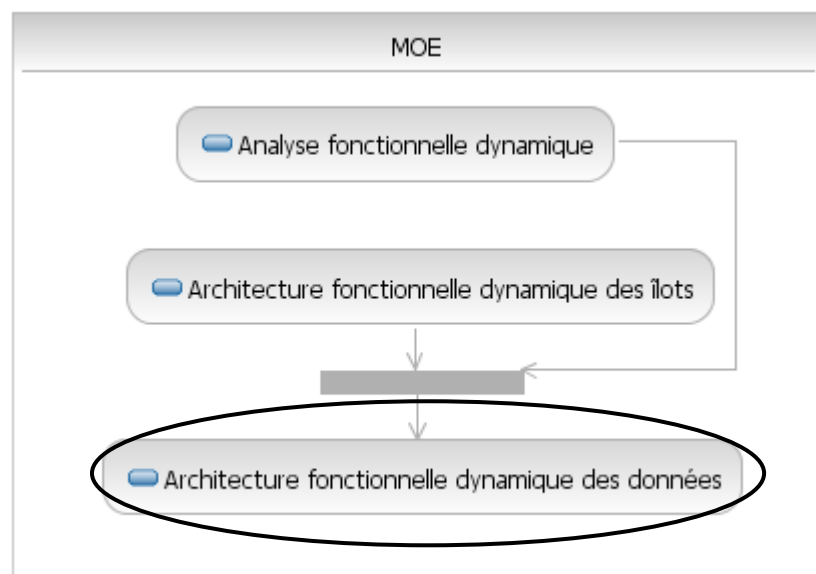


Figure 88 : Activité d'architecture fonctionnelle dynamique des données

Le premier concept obtenu lors de l'activité d'architecture fonctionnelle dynamique des données est l'attribut d'une donnée logique. L'attribut est en effet un paramètre d'entrée ou de sortie d'une ou plusieurs interactions entre instances d'îlots fonctionnels. Le second concept déduit immédiatement des attributs est la donnée logique regroupant des attributs. Le dernier concept est celui de voie entre îlots fonctionnels dont la conformité aux relations de dépendance entre données est stipulée dans la Règle d'urbanisme – Relation de dépendance entre données (cf. §4.1.2).

L'architecture fonctionnelle dynamique des données est la transformation décrite dans la **Figure 89** des attributs d'entités et des paramètres d'entrée ou de sortie des interactions entre entités en un modèle de données logiques.

La règle d'urbanisme associée à l'activité d'architecture fonctionnelle dynamique des données assure la cohérence du modèle des données logiques avec le modèle des îlots fonctionnels.

La définition d'une relation de dépendance entre données logiques est déduite de la définition d'une relation de dépendance entre attributs de données logiques.

Définition : Deux attributs, A_1 et A_2 , de données logiques ont une relation de dépendance telle que A_1 dépende de A_2 .

- si A_2 est le paramètre d'entrée et A_1 est le paramètre de sortie d'une interaction de type requête entre instances d'îlots fonctionnels dans un diagramme de séquence UML, ou,
- si A_2 est le paramètre de sortie d'une interaction de type requête entre instances d'îlots fonctionnels succédant directement à une interaction de type requête entre instances d'îlots fonctionnels ayant pour paramètre de sortie A_1 dans un diagramme de séquence UML.

Définition : Deux données logiques sont en relation de dépendance si au moins un attribut d'une des données logiques est en relation de dépendance avec au moins un attribut de l'autre donnée logique.

La Règle d'urbanisme – Architecture fonctionnelle dynamique des données permet de vérifier la cohérence du modèle de données avec le modèle des îlots fonctionnels.

Règle d'urbanisme – Architecture fonctionnelle dynamique des données n°13

- Tous les attributs d'une même donnée logique sont produits par un seul îlot fonctionnel conformément à la Règle d'urbanisme – Donnée produite et îlot fonctionnel décrite dans le §2.2.1.2 ;
- Deux attributs en relation de dépendance et produits par le même îlot fonctionnel peuvent être regroupés dans une même donnée logique,
- Chaque relation de dépendance entre données logiques est conforme à une voie entre îlots fonctionnels (cf. Règle d'urbanisme – Relation de dépendance entre données du §4.1.2).

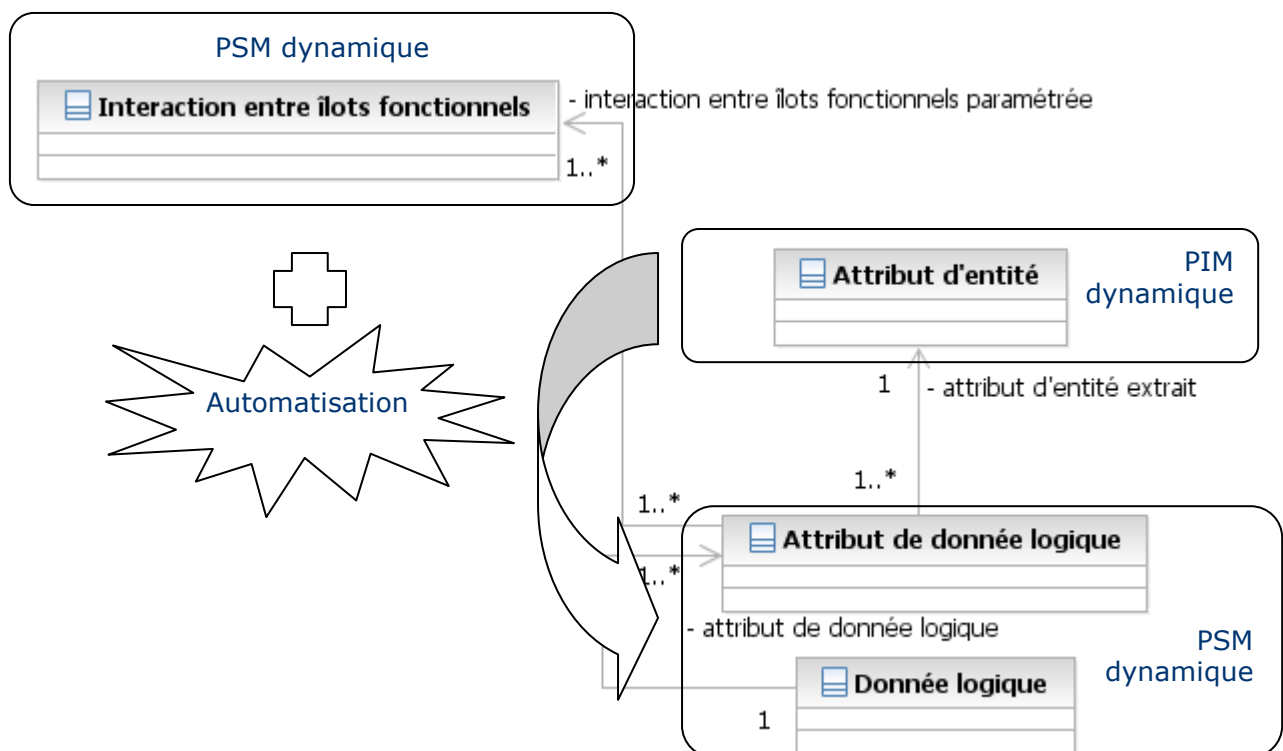


Figure 89 : Concepts et transformation liés à l'activité d'architecture fonctionnelle dynamique des données

Le mode de représentation des données logiques est le diagramme de classes UML stéréotypées <<donnée logique>> et le diagramme de séquence UML des interactions entre instances d'îlots fonctionnels où les données logiques et leurs attributs représentent les paramètres d'entrée ou de sortie des interactions. La donnée logique et son attribut sont séparés par le caractère "/".

Le mode de représentation des voies fonctionnelles reste celui des dépendances stéréotypées <<dépend de>> dans le diagramme de classes UML des îlots fonctionnels décrit dans le §3.2.

Le diagramme de séquence des interactions entre instances d'îlots fonctionnels représentant le scénario *SC-Filtrage négatif* de la **Figure 86** est complété par les paramètres d'entrée et de sortie. Le diagramme de séquence complet est décrit dans la **Figure 90**. Les paramètres des interactions de type requête sont les attributs des données logiques qui résultent de la transformation des paramètres d'entrée et de sortie des interactions entre entités représentés dans la **Figure 72**.

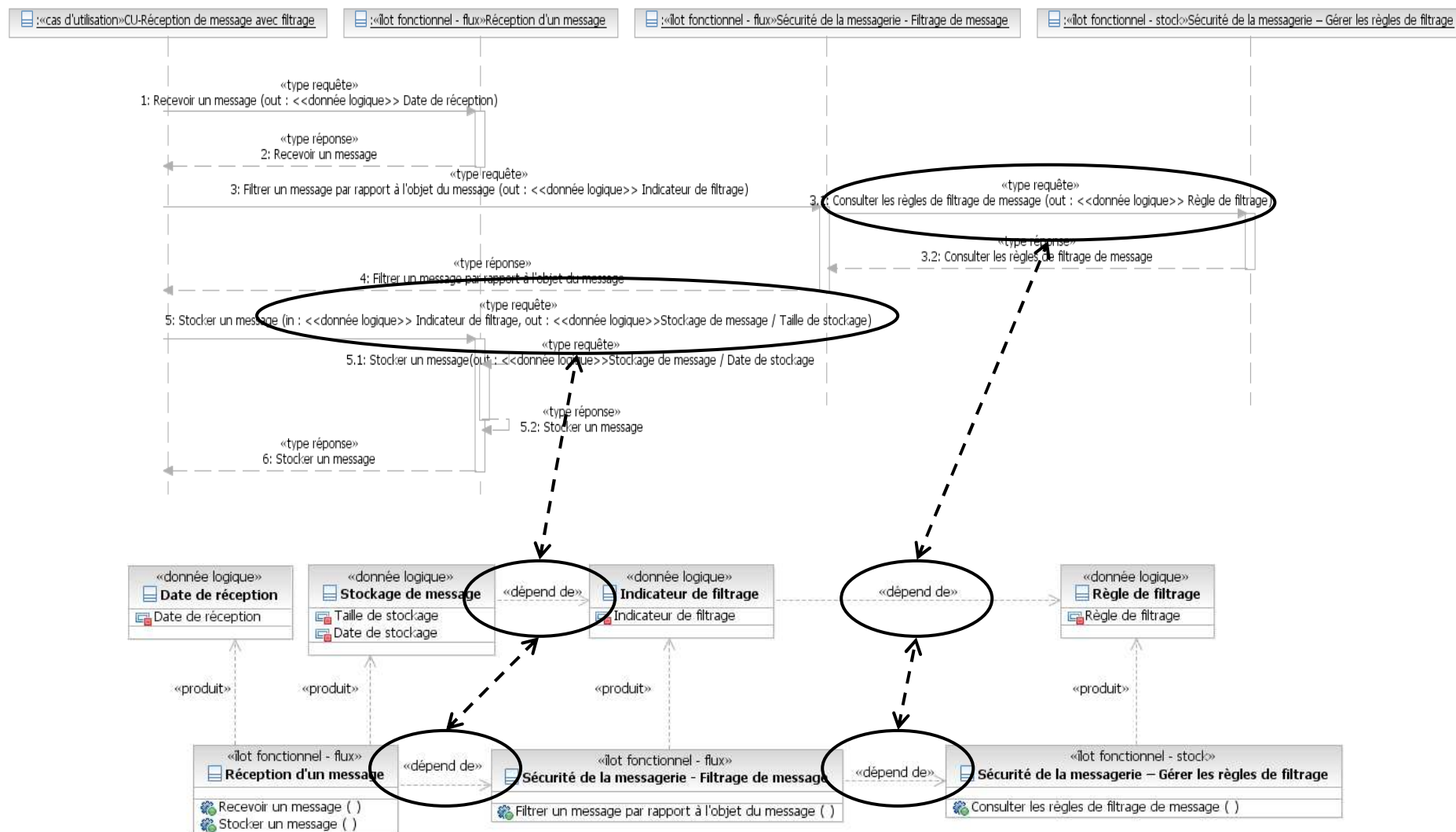


Figure 90 : Illustration des données logiques de l'activité d'architecture fonctionnelle dynamique des données

Le modèle de données logiques résultant de la transformation est constitué de :

- la donnée logique *Date de réception* définie par l'attribut *Date de réception*,
- la donnée logique *Indicateur de filtrage* définie par l'attribut *Indicateur de filtrage*,
- la donnée logique *Règle de filtrage* définie par l'attribut *Règle de filtrage*,
- la donnée logique *Stockage de message* définie par les attributs *Taille de stockage* et *Date de stockage*.

L'illustration de la de la **Figure 90** souligne le respect de la Règle d'urbanisme – Architecture fonctionnelle dynamique des données. La production d'une donnée logique par un îlot fonctionnel est représentée par un lien de dépendance stéréotypé <<produit>>.

- l'attribut *Date de réception* est produit par l'îlot fonctionnel *Réception d'un message*,
- l'attribut *Indicateur de filtrage* est produit par l'îlot fonctionnel *Sécurité de la messagerie – Filtrage de message*,
- l'attribut *Règle de filtrage* est produit par l'îlot fonctionnel *Sécurité de la messagerie – Gérer les règles de filtrage*,
- les attributs *Taille de stockage* et *Date de stockage* sont produits par l'îlot fonctionnel *Réception d'un message*,
- les attributs *Taille de stockage* et *Date de stockage* qui ont une relation de dépendance dans le diagramme de séquence représentant le scénario *SC-Filtrage négatif* sont regroupés dans la donnée logique *Stockage de message*,
- la relation de dépendance de la donnée logique *Stockage de message* vers la donnée logique *Indicateur de filtrage* est conforme à la voie de l'îlot fonctionnel *Réception d'un message* vers l'îlot fonctionnel *Sécurité de la messagerie – Filtrage de message* (relations entourées et reliées dans l'illustration),
- la relation de dépendance de la donnée logique *Indicateur de filtrage* vers la donnée logique *Règle de filtrage* est conforme à la voie de l'îlot fonctionnel *Sécurité de la messagerie – Filtrage de message* vers l'îlot fonctionnel *Sécurité de la messagerie – Gérer les règles de filtrage* (relations entourées et reliées dans l'illustration).

4.2.4 Conception de l'architecture technique

La conception de l'architecture technique du système est une macro-activité sous la responsabilité de l'équipe de développement, soit la MOE du système. Cette macro-activité, entourée dans la **Figure 91**, nécessite que le système soit analysé et que l'architecture fonctionnelle soit conçue. De plus, les préconisations faites par les urbanistes techniques contraignent cette activité. Comme pour l'urbanisme fonctionnel et son PLU, l'objectif est de sélectionner suivant des critères de performance et des critères économiques les nœuds d'exécution et les protocoles de la vue technique (cf. §2.2.1.2) à utiliser lors du développement d'un système du SI. L'activité d'urbanisme technique n'est pas détaillée ici méthodologiquement. L'évènement de validation des préconisations techniques est néanmoins intégré à la description des activités d'architecture technique.

Les nœuds d'exécution décrivant la vue technique d'éléments à réutiliser du SI sont étudiés lors de la macro-activité de conception de l'architecture technique. Cette étude permet de s'assurer de la viabilité technique de l'élément à réutiliser tel qu'il est détecté lors de l'activité d'architecture fonctionnelle statique.

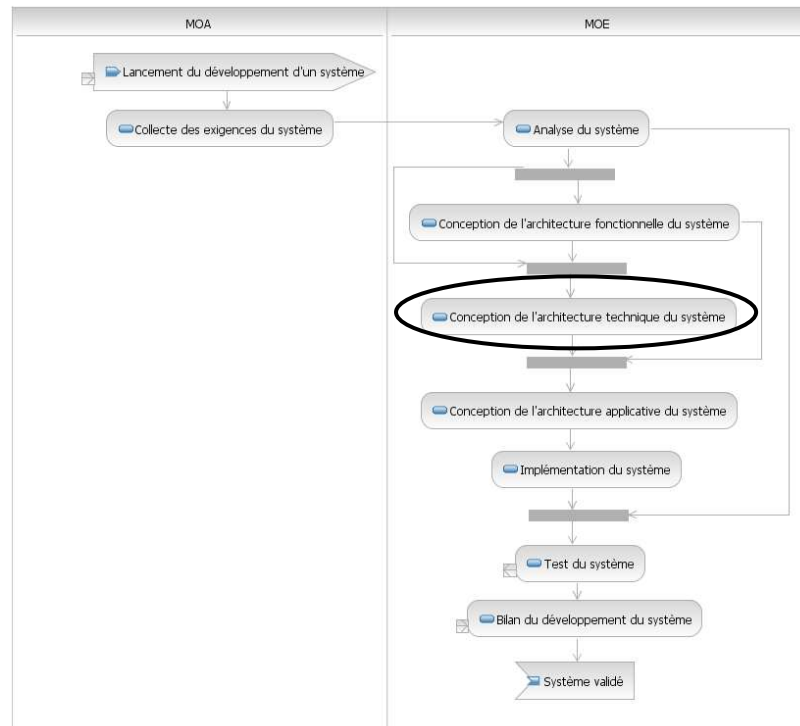


Figure 91 : Macro-activité de conception de l'architecture technique dans la démarche EA4UP

4.2.4.1 Architecture technique statique

L'activité de conception de l'architecture technique statique d'un système, entourée dans la **Figure 92**, a pour objectif de concevoir les nœuds d'exécution satisfaisant les propriétés techniques statiques du système spécifiées en analyse technique statique (cf. §4.2.2.3). Cette conception des nœuds d'exécution doit a priori respecter les préconisations de l'urbanisme technique. Une préconisation de nœud d'exécution est celle d'un type de nœud, par exemple, un nœud d'exécution de type serveur IBM x335 ou un OS Windows 200 AS SP3. Une préconisation d'un type de serveur ou d'un type d'OS (Operating System) permet de diminuer le coût d'achat auprès des fournisseurs, mais aussi de proposer une assistance interne performante aux utilisateurs des nœuds d'exécutions recommandés. Le nœud d'exécution peut constituer aussi la vue technique d'un élément réutilisable du SI détecté lors de l'activité d'architecture fonctionnelle statique (cf. §4.2.3.1).

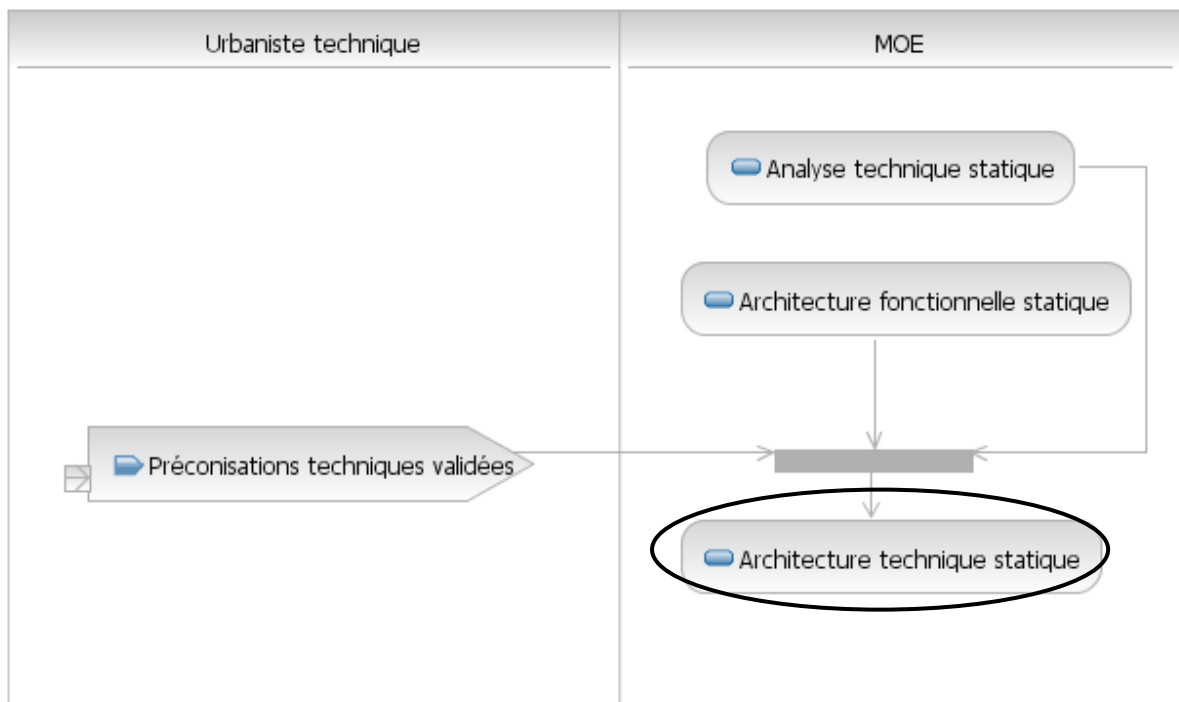


Figure 92 : Activité d'architecture technique statique

L'expert sélectionne parmi les nœuds d'exécution, dont le type est préconisé par l'urbaniste technique, ceux qui satisfont les propriétés techniques statiques du système (cf. **Figure 93**). Lorsque les propriétés techniques statiques l'exigent, un type de nœud d'exécution non préconisé peut être choisi par l'architecte technique du système. Les nœuds d'exécution du système sont cohérents avec certains îlots fonctionnels lorsque ceux-ci décrivent la vue fonctionnelle d'un élément à réutiliser du SI.

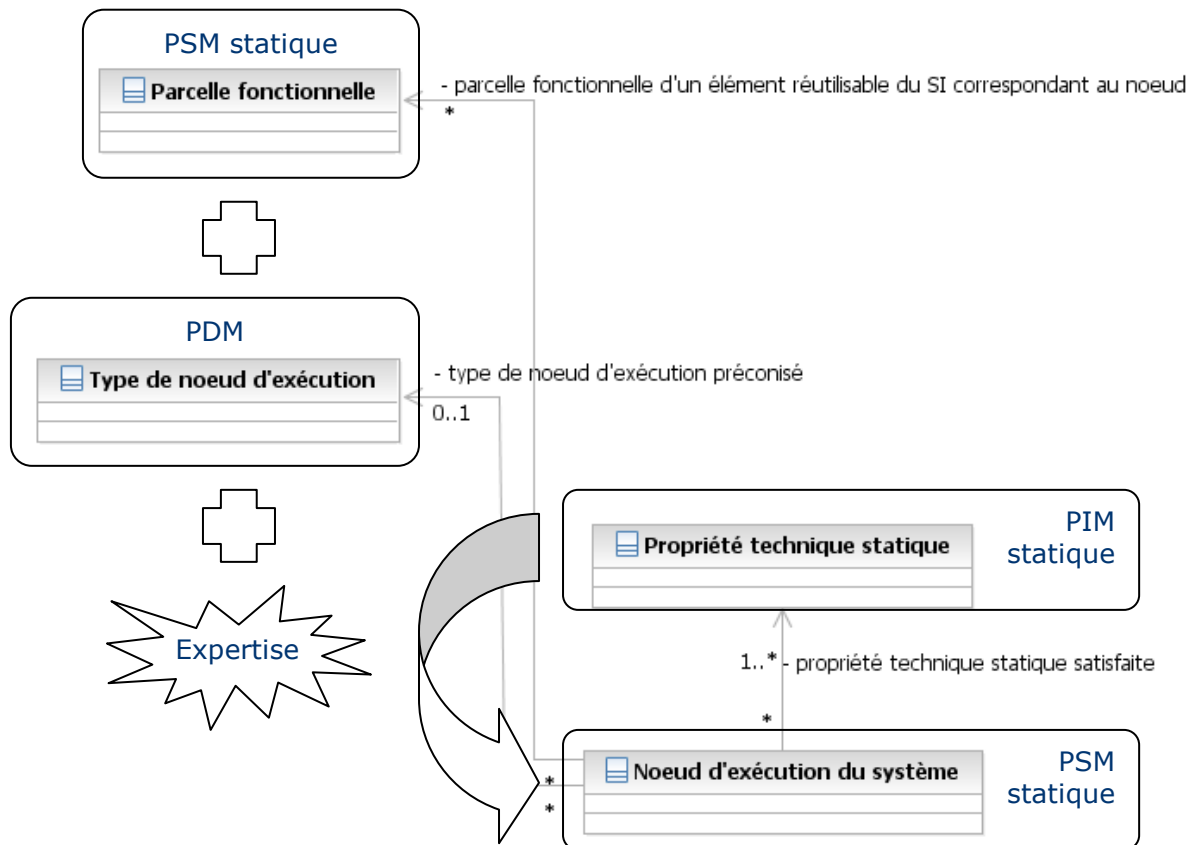


Figure 93 : Concepts et transformation liés à l'activité d'architecture technique statique

Un mode de représentation des nœuds d'exécution est celui des nœuds du diagramme de déploiement UML.

Dans le cas particulier d'un service télécom, un nœud d'exécution peut être la vue technique d'un enabler tels que défini par l'OMA (Open Mobile Alliance) [OMA, 2005]. Un enabler a en effet pour vocation d'être réutilisé lors de la conception de services télécom. Les éléments techniques composant la vue technique d'un enabler répondent à cette exigence. Pour les propriétés techniques statiques décrites dans le Tableau 12, le choix de l'architecte technique représenté dans la **Figure 94** confirme le résultat de l'architecture fonctionnelle statique quant aux enablers réutilisables :

- l'*Enabler Messagerie* qui satisfait l'association de l'exigence non fonctionnelle du nombre moyen de messages reçus *ENF-MMR* avec les attributs d'entité *Date de réception* et *Indicateur de filtrage*,
- l'*Enabler Stockage* qui satisfait l'association de l'exigence non fonctionnelle de la taille maximum de stockage *ENF-TMS* avec l'attribut d'entité *Taille de stockage* et le lien entre entités *est stocké dans*.

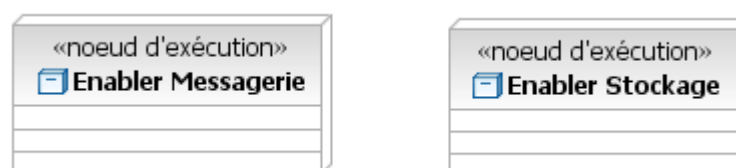


Figure 94 : Illustration des nœuds d'exécution de l'activité d'architecture technique statique

4.2.4.2 Architecture technique dynamique

L'activité de conception de l'architecture technique dynamique d'un système, entourée dans la **Figure 95**, a pour objectif de concevoir les protocoles satisfaisant les propriétés techniques dynamiques du système. Ces propriétés sont spécifiées en analyse technique dynamique (cf. §4.2.2.4). La conception des protocoles reliant les nœuds d'exécution conçus lors de l'activité d'architecture technique statique (cf. §4.2.4.1) doit respecter les préconisations de l'urbanisme technique. Une préconisation de protocole est, par exemple, un protocole de type FTP ou de type SIP. Ce choix commun d'un type de protocole pour le développement de systèmes d'un même SI permet d'assurer la cohérence globale des liens de communication entre nœuds d'exécution du SI.

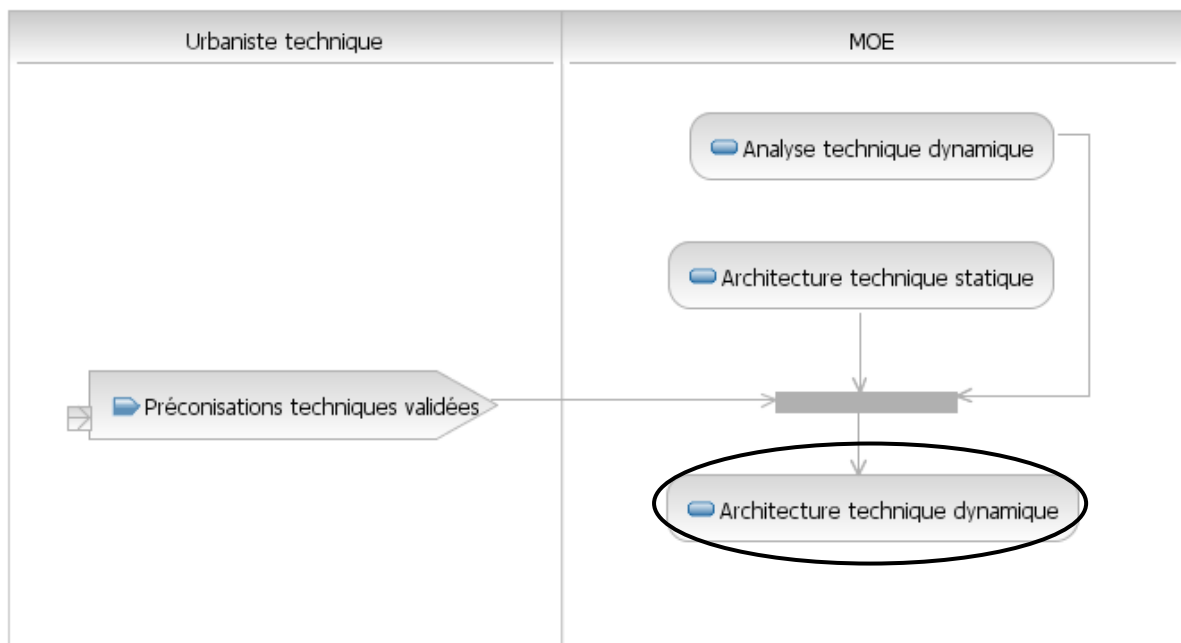


Figure 95 : Activité d'architecture technique dynamique

L'expert sélectionne parmi les protocoles, dont le type est préconisé par l'urbanisme technique, ceux qui satisfont les propriétés techniques dynamiques du système (cf. **Figure 96**). Les protocoles sélectionnés doivent être compatibles avec les nœuds d'exécution, conçus lors de l'activité d'architecture technique statique, qui les relient. Lorsque les propriétés techniques dynamiques l'exigent, un type de protocole non préconisé peut être choisi par l'architecte technique du système.

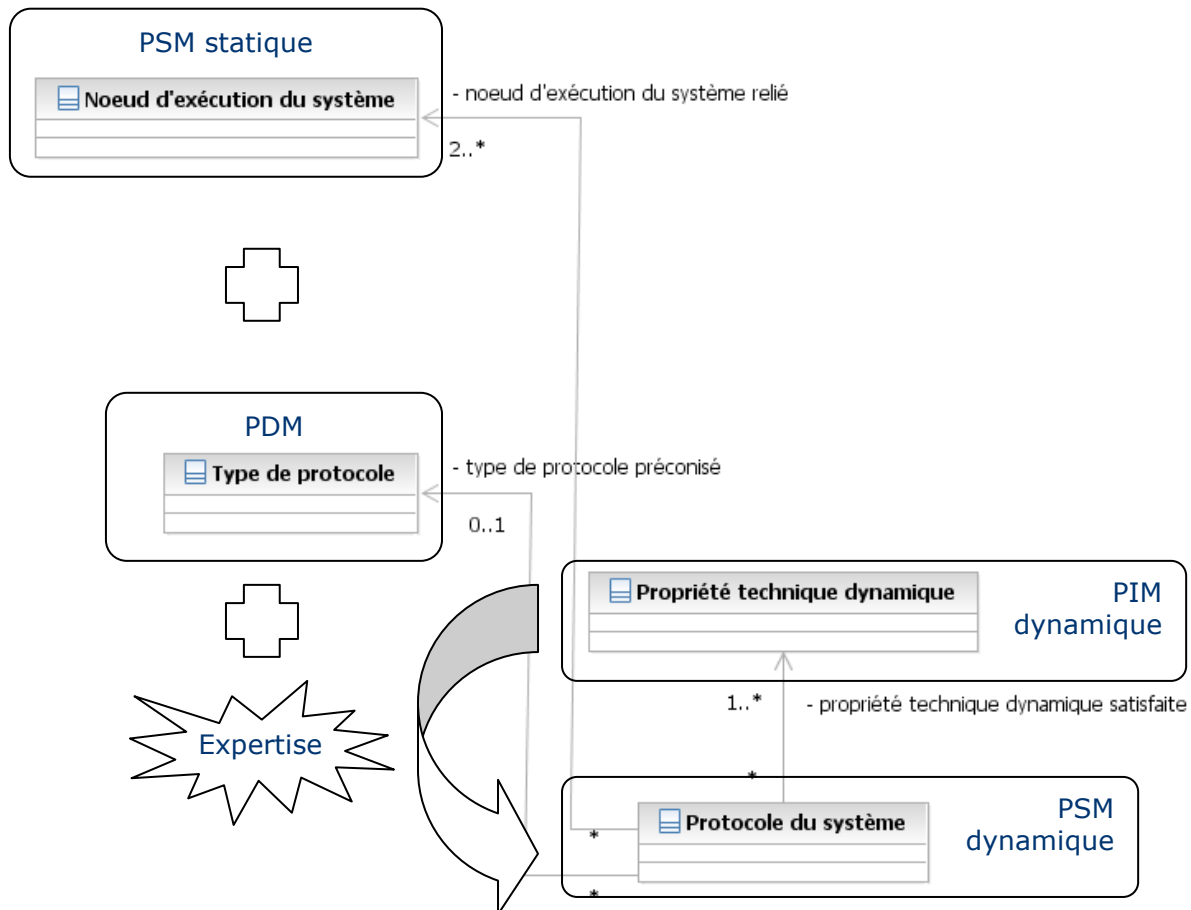


Figure 96 : Concepts et transformation liés à l'activité d'architecture technique dynamique

Un mode de représentation des protocoles est celui des liens de communication des diagrammes de déploiement UML.

Pour les propriétés techniques dynamiques décrites dans le diagramme de séquence de la **Figure 77**, le choix de l'architecte technique représenté dans la **Figure 97** s'est porté sur

- Le protocole *FTP* qui satisfait l'exigence non fonctionnelle ENF-TMS associée à l'interaction de type requête *Stocker un message* ayant pour source l'entité *Boîte de messagerie* associée au nœud d'exécution *Enabler Stockage* et pour cible l'entité *Message* associée au nœud d'exécution *Enabler Messagerie*.

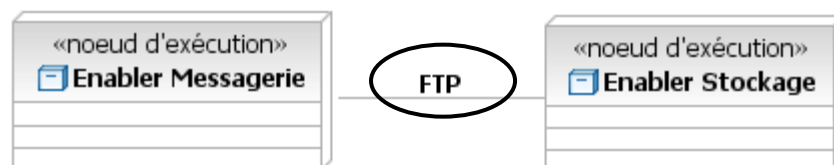


Figure 97 : Illustration d'un protocole de l'activité d'architecture technique dynamique

4.2.4.3 Architecture technique détaillée

L'activité d'architecture technique détaillée, entourée dans la **Figure 98**, permet de détailler les nœuds d'exécution conçus lors de l'activité d'architecture technique statique (cf. §4.2.4.1) ainsi que les protocoles conçus lors de l'activité d'architecture technique dynamique (cf. §4.2.4.2).

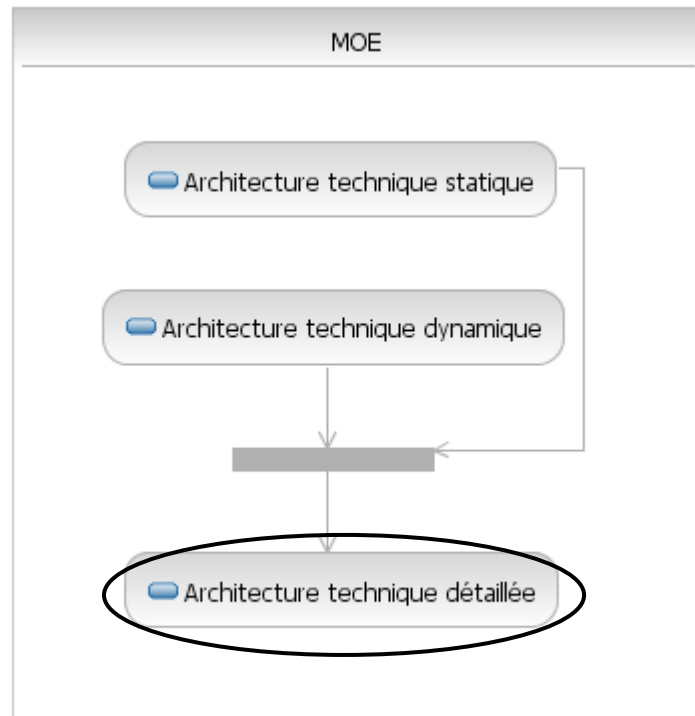


Figure 98 : Activité d'architecture technique détaillée

L'activité d'architecture technique détaillée n'est pas explicitée ici car elle ne comporte pas de spécificité dans la démarche EA4UP.

4.2.5 Conception de l'architecture applicative

La conception de l'architecture applicative du système est une macro-activité sous la responsabilité de l'équipe de développement, soit la MOE du système. Cette macro-activité, entourée dans la **Figure 99**, nécessite que le système soit fonctionnellement et techniquement architecturé. L'objectif de l'architecture applicative est la conception de composant applicatifs et de leurs interfaces réalisant l'architecture fonctionnelle du système (cf. §4.2.3) et étant déployée sur l'architecture technique du système (cf. §4.2.4).



Figure 99 : Macro-activité de conception de l'architecture applicative dans la démarche EA4UP

La transformation pour réaliser les activités de la macro-activité d'architecture applicative est plus classique par rapport à l'approche MDA. En effet, l'expert transforme un modèle d'architecture fonctionnelle en un modèle d'architecture applicative en respectant une contrainte représentée par un modèle d'architecture technique. Le PDM modélisant les éléments applicatifs à réutiliser dans le SI est ajouté à cette contrainte.

4.2.5.1 Architecture applicative statique

L'activité de conception de l'architecture applicative statique d'un système, entourée dans la **Figure 100**, a pour objectif de concevoir les composants applicatifs qui réalisent les îlots conçus lors de l'architecture fonctionnelle statique (cf. §4.2.3.1). Les composants applicatifs sont déployés sur les nœuds d'exécution conçus lors de l'activité d'architecture technique statique (cf. §4.2.4.1). La validation d'éléments applicatifs réutilisables est une préconisation faite pour un SI par un urbaniste applicatif, expert de la vue applicative du SI.



Figure 100 : Activité d'architecture applicative statique

L'expert transforme l'ensemble des parcelles fonctionnelles définissant les îlots du système en un ensemble de composants applicatifs. Chaque composant applicatif représenté dans la **Figure 101** est alors déployé sur un nœud d'exécution. La Règle d'urbanisme – Architecture applicative statique assure la cohérence entre les composants applicatifs, les parcelles fonctionnelles réalisées et les nœuds d'exécution permettant le déploiement de ces composants. Dans le cas particulier où le nœud d'exécution est un élément réutilisable du SI, les composants applicatifs sont ceux définis dans la vue applicative de cet élément.

Règle d'urbanisme – Architecture applicative statique n°14 :

- Chaque instance de parcelle fonctionnelle ne peut être réalisée que par un seul composant applicatif ;
- Chaque composant applicatif réalise, soit des parcelles fonctionnelles d'îlots de type flux, soit des parcelles fonctionnelles d'îlots de type stock ;
- Chaque composant applicatif est déployé sur un seul nœud d'exécution ;
- Lorsque le nœud de déploiement caractérise un élément du SI réutilisé, les composants applicatifs déployés sur ce nœud sont conformes à cet élément réutilisé, en particulier, à la définition des composants applicatifs qui composent sa vue applicative.

Une illustration du premier item de la Règle d'urbanisme – Architecture applicative statique est que l'instanciation de la parcelle fonctionnelle *Recevoir un message* instanciée pour un message électronique ne peut être réalisée que par un seul composant applicatif.

L'intérêt du second item est de séparer la réalisation d'îlots de type flux de la réalisation d'îlots de type stock. Cette séparation permet de caractériser les échanges entre composants applicatifs. En effet, les requêtes envoyées vers un composant applicatif réalisant du flux sont corrélées à chaque usage du service alors que les requêtes envoyées vers un composant applicatif réalisant du stock sont décorrélées de l'usage du service. Le lien de communication de la vue technique supportant ces échanges est ainsi rendu spécifique au typage de ce que réalise chaque composant cible d'une interaction de type requête.

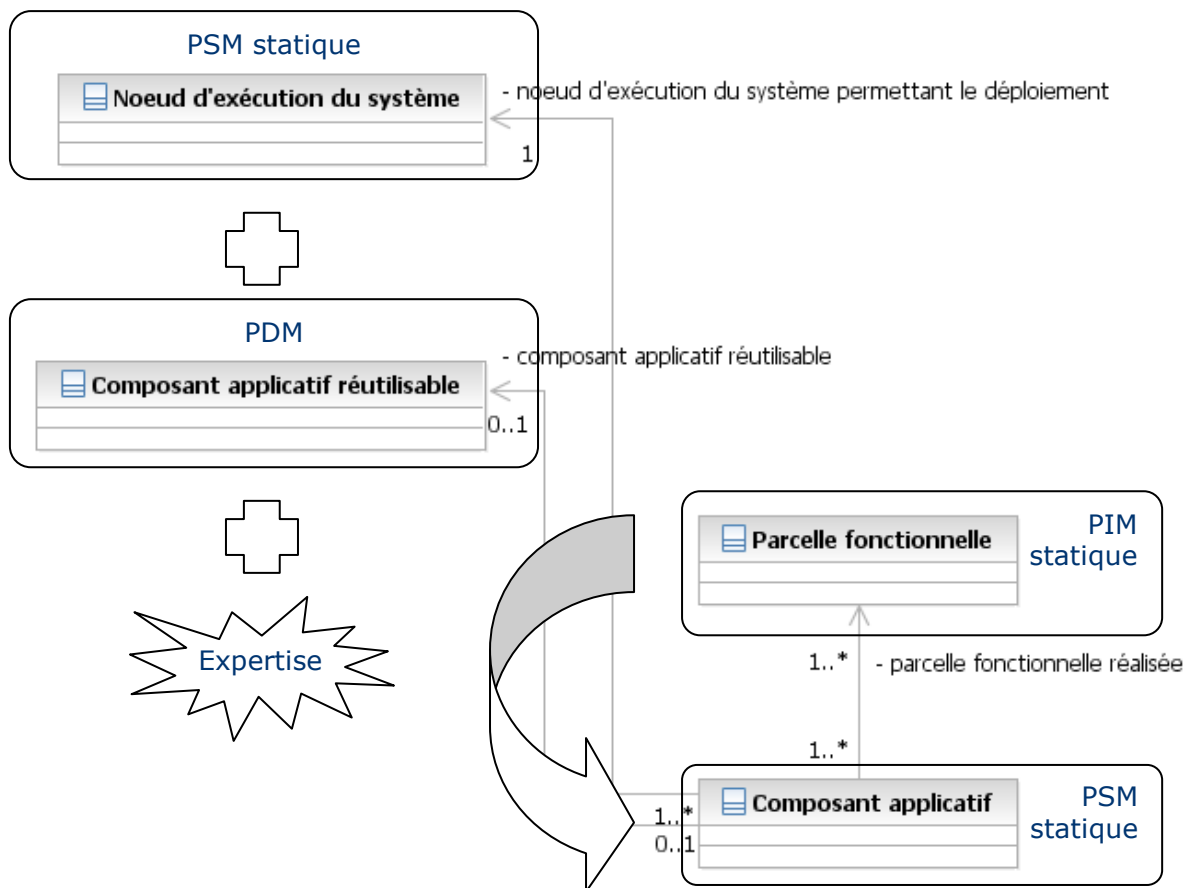


Figure 101 : Concepts et transformation liés à l'activité d'architecture applicative statique

Un mode de représentation des composants applicatifs est un diagramme de composants UML où chaque composant est stéréotypé <<composant applicatif>>.

Le choix de l'architecte applicatif du système est la conception de trois composants représentés dans la **Figure 102** et appartenant à la vue applicative des enablers réutilisés pour ce développement :

- *Messagerie,*
- *Stockage,*
- *Sécurité.*



Figure 102 : Illustration de composants applicatifs de l'activité d'architecture applicative statique

Le Tableau 14 représente les parcelles des îlots fonctionnels réalisées par chaque composant applicatif.

Composant applicatif	Parcelle fonctionnelle réalisée	Îlot fonctionnel
Messagerie	<i>Recevoir un message</i>	Réception d'un message
	<i>Filtrer un message par rapport à l'objet du message</i>	Sécurité de la messagerie – Filtrage de message
Stockage	<i>Stocker un message</i>	Réception d'un message
Filtrage	<i>Consulter une règle de filtrage de message</i>	Sécurité de la messagerie – Gérer les règles de filtrage

Tableau 14 : Illustration du lien de réalisation d'une parcelle fonctionnelle par un composant applicatif de l'activité d'architecture applicative statique.

La Règle d'urbanisme – Architecture applicative statique (cf. §4.2.5.1) est vérifiée pour le lien de réalisation avec les parcelles fonctionnelles puisque :

- la seule instance est celle d'un message électronique,
- le composant applicatif *Messagerie* ne réalise que des parcelles d'îlots fonctionnels de type flux,
- le composant applicatif *Stockage* ne réalise qu'une parcelle d'îlot fonctionnel de type flux,
- le composant applicatif *Filtrage* ne réalise qu'une parcelle d'îlot fonctionnel de type stock.

Le diagramme de déploiement UML de la **Figure 103** représente le déploiement de chaque composant applicatif sur un nœud d'exécution.

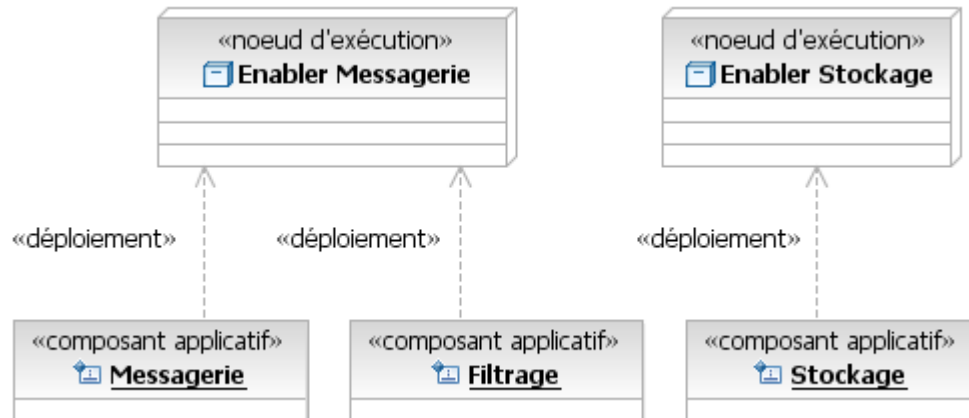


Figure 103 : Illustration du lien de déploiement d'un composant applicatif de l'activité d'architecture applicative statique sur un nœud d'exécution

La Règle d'urbanisme – Architecture applicative statique est vérifiée pour le lien de déploiement de chaque composant applicatif sur un nœud d'exécution et pour la réutilisation des enablers :

- les composants applicatifs *Messagerie* et *Filtrage* sont déployés sur le nœud d'exécution *Enabler Messagerie*,
- le composant applicatif *Stockage* est déployé sur le nœud d'exécution *Enabler Stockage*,
- les composants applicatifs *Messagerie* et *Filtrage* sont ceux définis dans la vue applicative de l'*Enabler Messagerie* réutilisé,
- le composant applicatif *Stockage* est celui défini dans la vue applicative de l'*Enabler Stockage* réutilisé.

4.2.5.2 Architecture applicative dynamique des composants

L'activité de conception de l'architecture applicative dynamique des composants d'un système, entourée dans la **Figure 104**, a pour objectif de concevoir les interactions entre composants applicatifs conçus lors de l'activité d'architecture applicative statique (cf. §4.2.5.1). Les interactions entre composants applicatifs réalisent les interactions entre îlots fonctionnels décrites lors de l'activité d'architecture fonctionnelle dynamique des îlots (cf. §4.2.3.2). Cette conception des interactions entre îlots fonctionnels est contrainte par l'existence de protocoles conçus lors de l'activité d'architecture technique dynamique (cf. §4.2.4.2). Comme pour l'activité précédente, les éléments applicatifs réutilisables, en particulier les interfaces applicatives, constituent une préconisation d'urbanisme applicatif du SI. Cette préconisation est à prendre en compte lors de l'activité d'architecture applicative dynamique des composants.

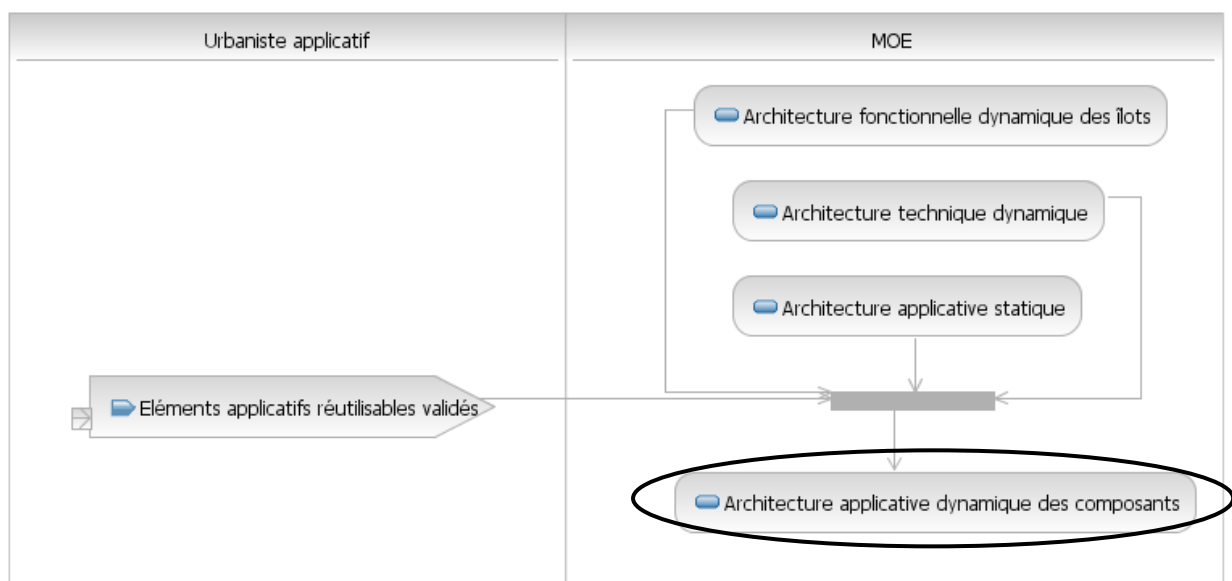


Figure 104 : Activité d'architecture applicative dynamique des composants

L'activité d'architecture applicative dynamique des composants applicatifs est la transformation du modèle des interactions conçues entre les îlots de la vue fonctionnelle du système. Cette transformation représentée dans la **Figure 105** est contrainte par les composants applicatifs réalisant les îlots fonctionnels du système. L'existence de protocoles entre les nœuds d'exécution où sont déployés les composants applicatifs de l'interaction est aussi une contrainte de la transformation. L'automatisation de la transformation est rendue possible :

- par le lien entre les composants applicatifs de l'interaction et les îlots fonctionnels qu'ils réalisent et
- par le lien entre les composants applicatifs de l'interaction et les nœuds d'exécution permettant leur déploiement.

La Règle d'urbanisme – Architecture applicative dynamique des composants permet cette automatisation.

Règle d'urbanisme – Architecture applicative dynamique des composants n°15

- Chaque interaction entre îlots fonctionnels est réalisée par une seule interaction entre composants applicatifs ;
- Si l'interaction de type requête est adressée à une instance du composant applicatif C_2 à partir d'une instance du composant applicatif C_1 , elle nécessite l'utilisation par le composant applicatif C_1 d'une interface applicative fournie par le composant applicatif C_2 ;
- Chaque interaction entre composants applicatifs est déployée sur le protocole qui relie les nœuds d'exécution où sont déployés les composants applicatifs de l'interaction ;
- Lorsque l'un des deux nœuds de déploiement reliés par le protocole supportant l'interaction caractérise un élément du SI réutilisé, les interactions entre composants applicatifs déployés sur ce nœud doivent respecter la vue applicative de cet élément réutilisé, en particulier la définition des interfaces applicatives.

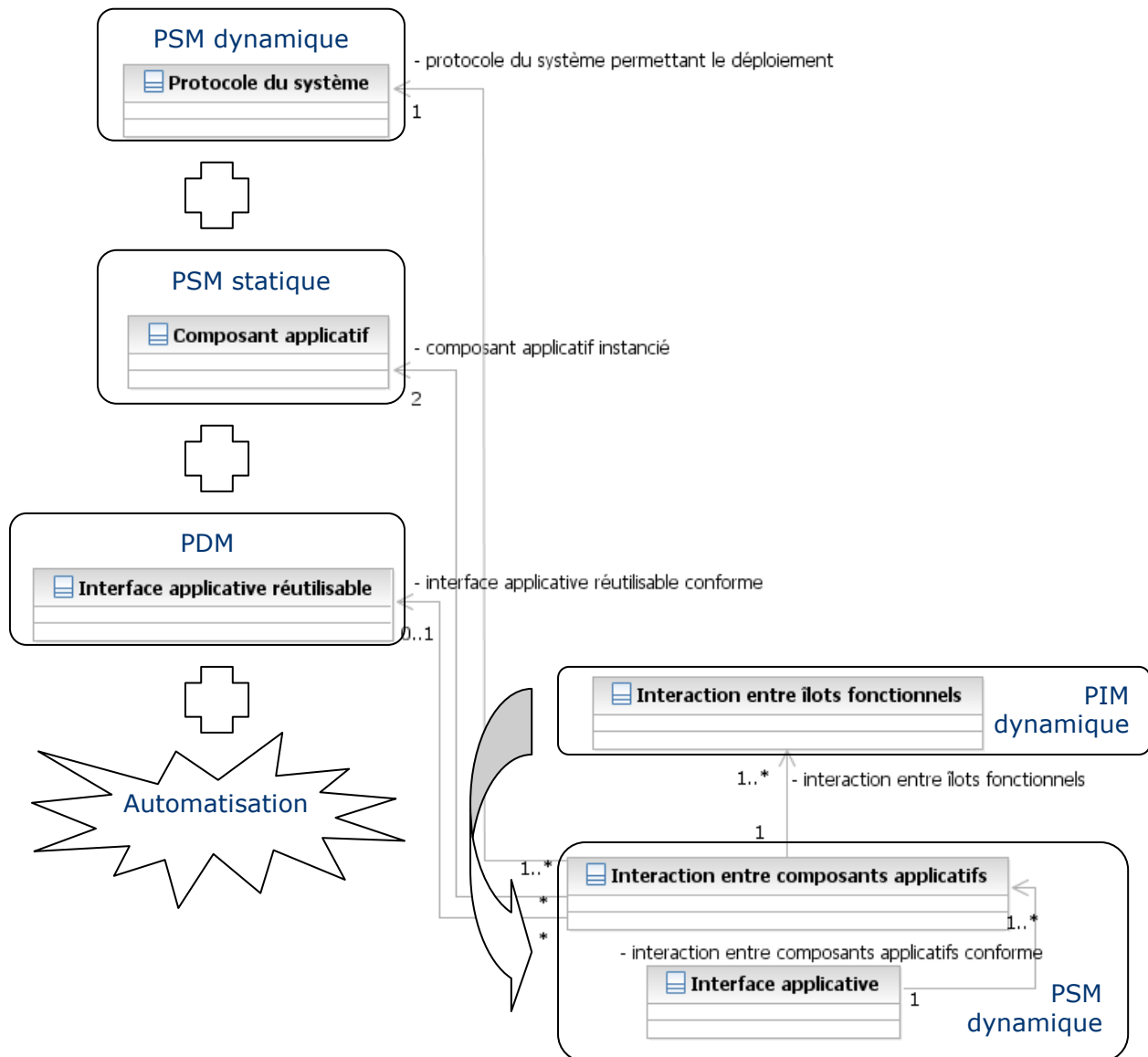


Figure 105 : Concepts et transformation liés à l'activité d'architecture applicative dynamique des composants

D'après le second item de la Règle d'urbanisme – Architecture applicative dynamique des composants (cf. §4.2.5.2) il est immédiat de déduire les interfaces applicatives fournies et utilisées par chaque composant applicatif à partir des interactions conçues entre composants applicatifs.

Un mode de représentation des interactions entre composants applicatifs est un diagramme de séquence UML représentant les instances de composants applicatifs réalisant un scénario d'utilisation du système. Chaque interaction est libellée par le nom de l'interface applicative utilisée lors de la requête. La représentation de la succession temporelle d'interactions et la représentation de la dépendance d'interactions respecte la Sémantique UML – Approche dynamique et diagramme de séquence (cf. §2.1.5).

Le mode de représentation des interfaces applicatives sont les interfaces utilisées et fournies dans un diagramme de composants UML.

Dans la **Figure 106**, le diagramme de séquence composé d'interactions entre instances de composants applicatifs représente le scénario *SC-Filtrage négatif*. Ce diagramme est le résultat de la transformation du diagramme de séquence composé d'interactions entre îlots fonctionnels de la **Figure 86**. Le déploiement des interactions entre instances de composants applicatifs sur le protocole FTP est schématisé par un cylindre sur le diagramme.

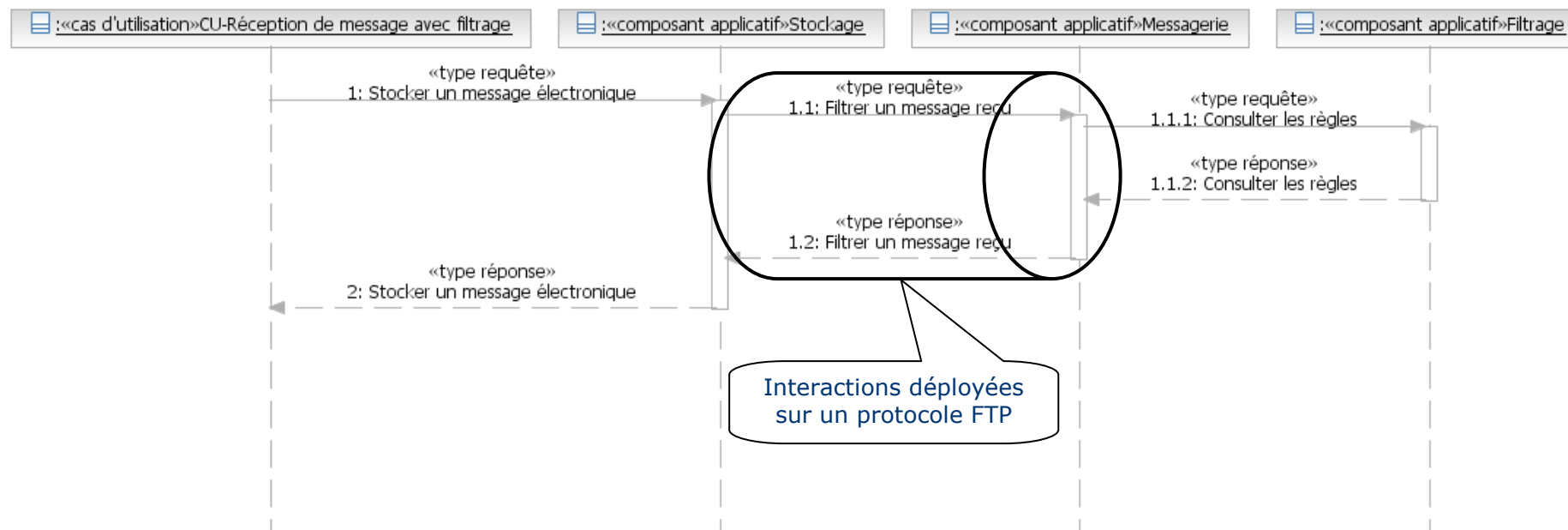


Figure 106 : Illustration des interactions entre composants applicatifs de l'activité d'architecture applicative dynamique des composants

Parmi les diagrammes de séquence automatiquement générés à partir du diagramme de séquence représentant des instances d'îlots fonctionnels réalisant *SC-Filtrage négatif*, le diagramme de séquence retenu est celui de l'enchaînement d'interactions ayant pour valeur ajoutée le stockage d'un message reçu après filtrage.

Le premier item de la Règle d'urbanisme – Architecture applicative dynamique des composants (cf. §4.2.5.2) est respecté :

- l'interaction entre îlots fonctionnels *Recevoir un message* est réalisée par l'interaction entre composants organiques *Filtrer un message reçu* ;
- l'interaction entre îlots fonctionnels *Filtrer un message par rapport à l'objet du message* est réalisée par l'interaction entre composants organiques *Filtrer un message reçu* ;
- l'interaction entre îlots fonctionnels *Consulter les règles de filtrage de message* est réalisée par l'interaction entre composants organiques *Consulter les règles* ;
- les deux interactions entre îlots fonctionnels *Stocker un message* sont réalisées par l'interaction entre composants organiques *Stocker un message électronique*.

La satisfaction du dernier item de la Règle d'urbanisme – Architecture applicative dynamique des composants est soulignée dans la **Figure 106** avec le déploiement de l'interaction *Filtrer un message reçu* sur le protocole FTP, sachant que :

- le composant applicatif *Stockage* est déployé sur le nœud d'exécution *Enabler Stockage*,
- le composant applicatif *Messagerie* est déployé sur le nœud d'exécution *Enabler Messagerie*,
- le protocole FTP relie les nœuds d'exécution *Enabler Stockage* et *Enabler Messagerie*.

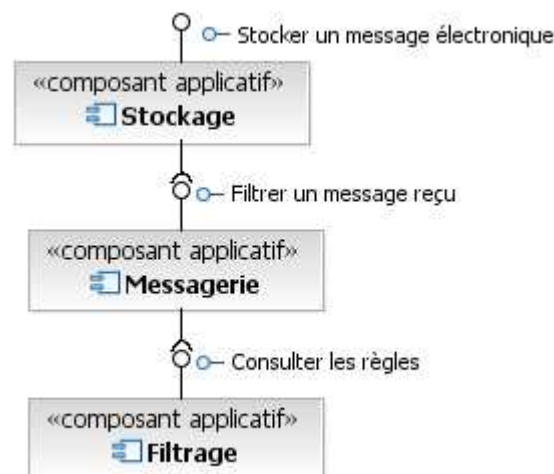


Figure 107 : Illustration des interfaces applicatives de l'activité d'architecture applicative dynamique des composants

Les interfaces applicatives décrites dans le diagramme de composants de la **Figure 107** respecte le deuxième item de la Règle d'urbanisme – Architecture applicative dynamique des composants. Les interfaces applicatives sont en effet cohérentes avec les interactions décrites dans le diagramme de séquence de la **Figure 106** :

- L'interface applicative *Stocker un message électronique* fournie par le composant *Stockage* correspond à l'interaction de requête, de même libellé, adressée du scénario à ce composant applicatif ;
- L'interface applicative *Filtrer un message reçu* fournie par le composant *Messagerie* et utilisée par le composant applicatif *Stockage* correspond à l'interaction de requête, de même libellé, entre ces deux composants applicatifs ;
- L'interface applicative *Consulter les règles* fournie par le composant *Filtrage* et utilisée par le composant applicatif *Messagerie* correspond à l'interaction de requête, de même libellé, entre ces deux composants applicatifs.

4.2.5.3 Architecture applicative dynamique des données

L'activité de conception de l'architecture applicative des données d'un système, entourée dans la **Figure 108**, a pour objectif de concevoir les attributs des données physiques. Ceux-ci sont les paramètres de sortie des interactions entre instances de composants applicatifs. Ces interactions sont conçues lors de l'activité d'architecture dynamique des composants applicatifs (cf. §4.2.5.2). Les attributs des données physiques réalisent des attributs de données logiques conçus lors de l'activité d'architecture fonctionnelle dynamique des données (cf. §4.2.3.3). Le respect de la Règle d'urbanisme – Relation de dépendance entre données (cf. §4.1.2) signifie que les interfaces applicatives fournies et utilisées conçues lors de l'activité d'architecture dynamique des composants applicatifs peuvent évoluer lors de l'activité d'architecture applicative dynamique des données.

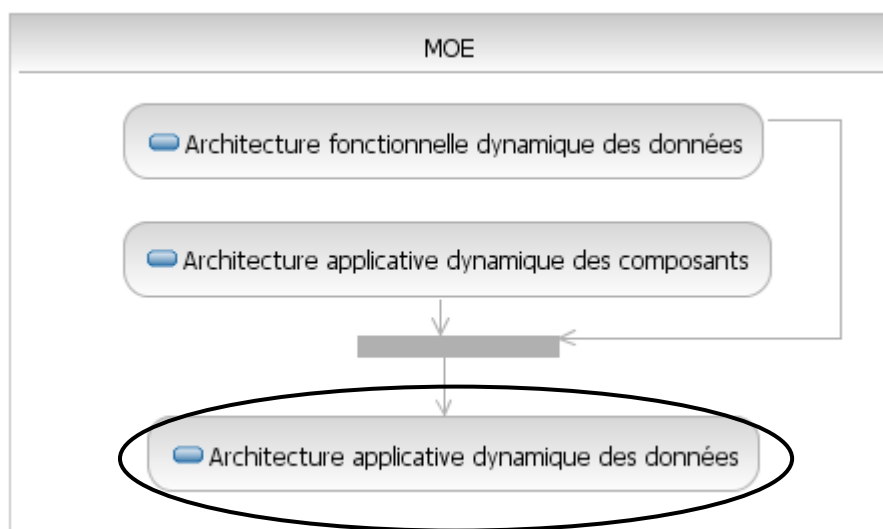


Figure 108 : Activité d'architecture applicative dynamique des données

Le premier concept obtenu lors de l'activité d'architecture applicative dynamique des données est l'attribut d'une donnée physique décrit dans la **Figure 109**. L'attribut est en effet un paramètre d'entrée ou de sortie d'une, ou plusieurs, interactions entre instances de composants applicatifs. Le deuxième concept déduit immédiatement des attributs sont les données physiques regroupant ces attributs. Le dernier concept est celui des interfaces applicatives fournies et utilisées par les composants applicatifs. La conformité des interfaces applicatives aux relations de dépendance entre données physiques est stipulée dans la Règle d'urbanisme – Relation de dépendance entre données (cf. §4.1.2).

L'architecture applicative dynamique des données est la transformation des attributs de données logiques et des paramètres d'entrée ou de sortie des interactions entre îlots fonctionnels en un modèle de données physiques.

La règle d'urbanisme associée à l'activité d'architecture applicative dynamique des données assure la cohérence du modèle de données physiques avec le modèle des composants applicatifs et de leurs interfaces.

La définition d'une relation de dépendance entre données physiques est déduite de celle des attributs de données physiques.

Définition : Deux attributs, A_1 et A_2 , de données physiques sont en relation de dépendance telle que A_1 dépende de A_2 .

- si A_2 est le paramètre d'entrée et A_1 est le paramètre de sortie d'une interaction de type requête entre instances de composants applicatifs dans un diagramme de séquence UML, ou,
- si A_2 est le paramètre de sortie d'une interaction de type requête entre instances de composants applicatifs succédant directement à une interaction de type requête entre instances de composants applicatifs ayant pour paramètre de sortie A_1 dans un diagramme de séquence UML.

Définition : Deux données physiques sont en relation de dépendance si au moins un attribut d'une des données physiques est en relation de dépendance avec au moins un attribut de l'autre donnée physique.

La Règle d'urbanisme – Architecture applicative dynamique des données permet de vérifier la cohérence du modèle de données physiques avec le modèle des composants applicatifs.

Règle d'urbanisme – Architecture applicative dynamique des données n°16

- Tous les attributs d'une même donnée physique sont produits par un seul composant applicatif conformément à la Règle d'urbanisme – Donnée produite et composant applicatif (cf. §2.2.1.2) ;
- Deux attributs, en relation de dépendance, produits par le même composant applicatif peuvent être regroupés dans une même donnée physique ;
- Chaque relation de dépendance entre données physiques est conforme à une interface applicative
 - o fournie par le composant applicatif producteur de la donnée cible de la dépendance et
 - o utilisée par le composant applicatif producteur de la donnée source de la dépendance ;
- Lorsque le nœud de déploiement du composant applicatif producteur d'une donnée physique caractérise un élément du SI réutilisé, la donnée physique produite par le composant applicatif déployé sur ce nœud doit respecter la vue applicative de cet élément réutilisé, en particulier la définition des données physiques produites.

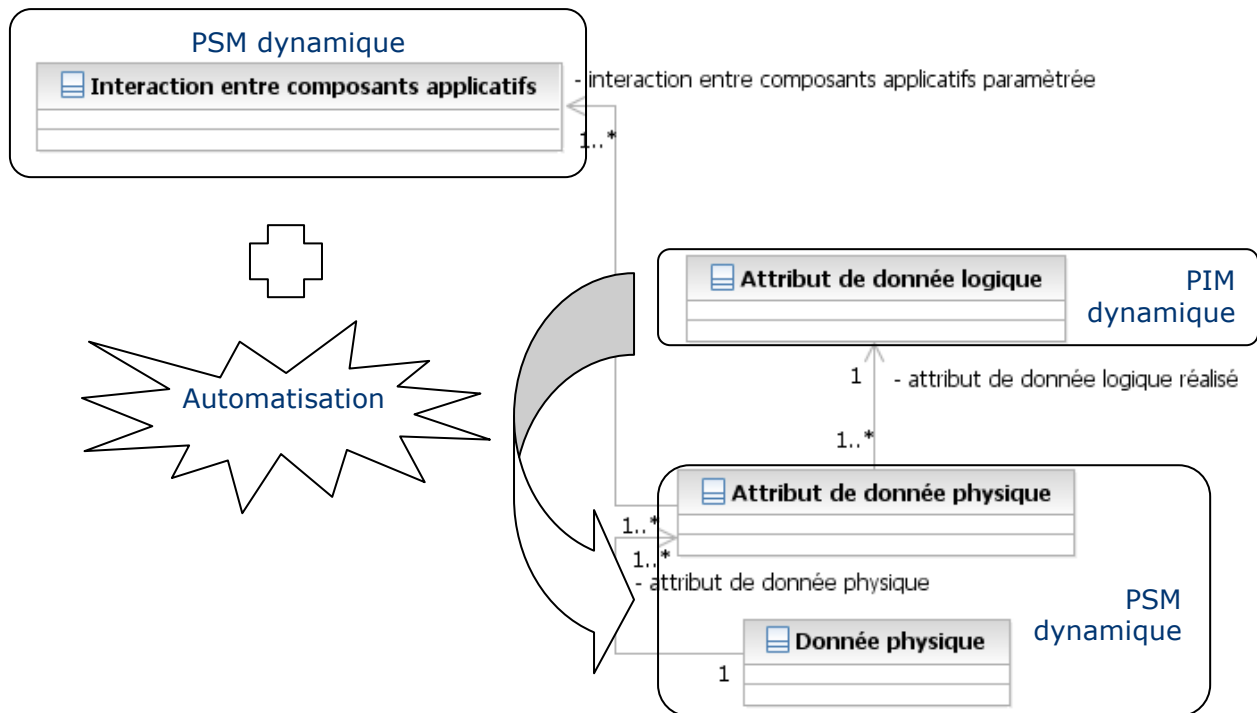


Figure 109 : Concepts et transformation liés à l'activité d'architecture applicative dynamique des données

Le mode de représentation des données physiques est le diagramme de classes UML stéréotypées <<donnée physique>>. Le diagramme de séquence UML représente les interactions entre instances de composants applicatifs. Les données physiques et leurs attributs sont les paramètres d'entrée ou de sortie des interactions. La donnée physique et son attribut sont séparés par le caractère "/".

Le mode de représentation des interfaces applicatives reste celui des interfaces dans un diagramme de composants UML.

Le diagramme de séquence des interactions entre instances de composants applicatifs représentant le scénario *SC-Filtrage négatif* de la **Figure 106** est complété par les paramètres d'entrée et de sortie dans la **Figure 110**. Les paramètres des interactions de type requête sont les attributs des données physiques résultant de la transformation des paramètres d'entrée et de sortie des interactions entre îlots fonctionnels représentés dans la **Figure 90**.

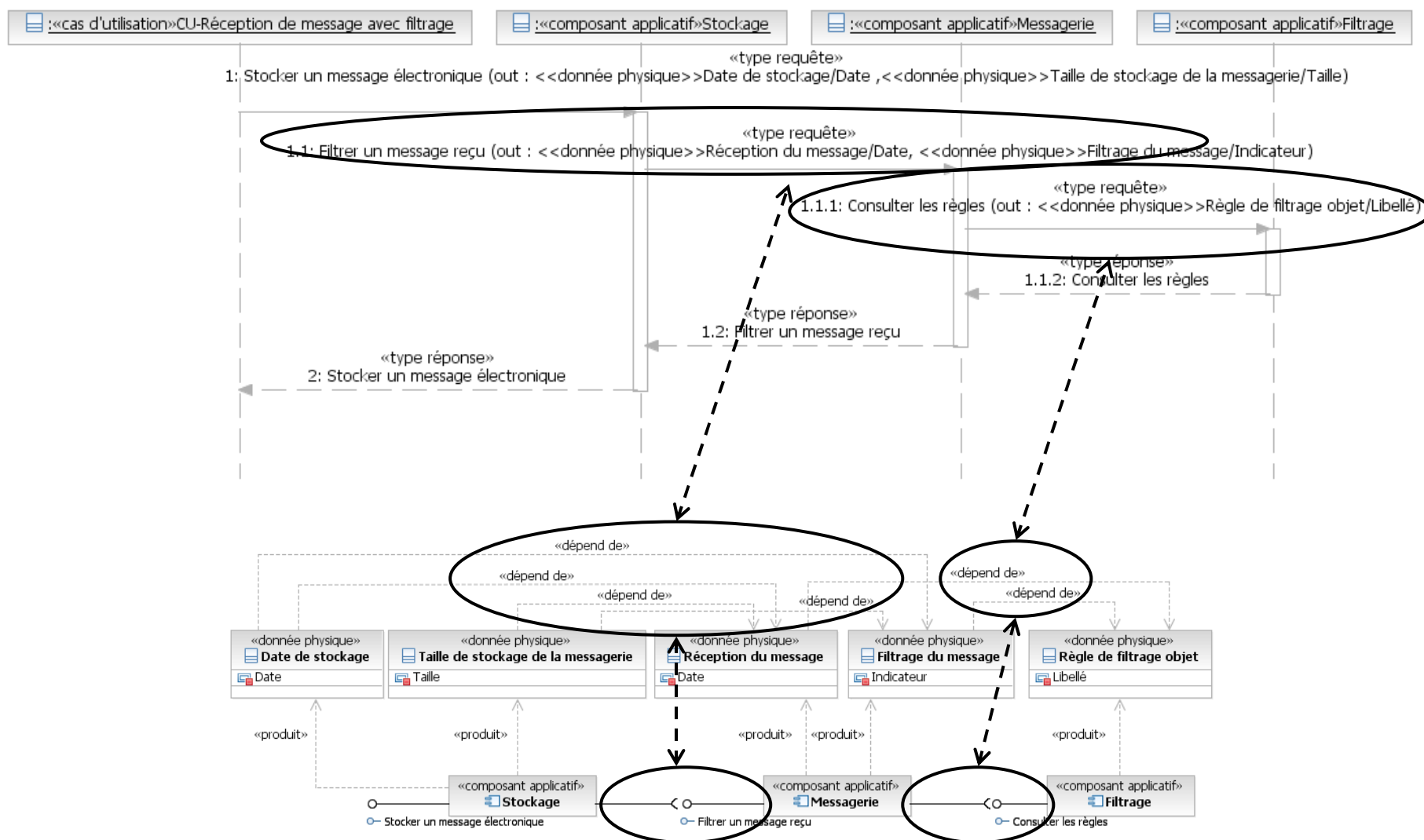


Figure 110 : Illustration des données physiques de l'activité d'architecture applicative dynamique des données

Le modèle de données physiques résultant de la transformation est constitué de :

- la donnée physique *Date de stockage* définie par l'attribut *Date*,
- la donnée physique *Taille de stockage de la messagerie* définie par l'attribut *Taille*,
- la donnée physique *Réception du message* définie par l'attribut *Date*,
- la donnée physique *Filtrage de message* définie par l'attribut *Indicateur*,
- la donnée physique *Règle de filtrage objet* définie par l'attribut *Libellé*.

L'illustration de la **Figure 110** souligne le respect de la Règle d'urbanisme – Architecture applicative dynamique des données. La production d'une donnée physique par un composant applicatif est représentée par un lien de dépendance stéréotypé <<produit>>.

- l'attribut *Date* de la donnée physique *Date de stockage* est produit par le composant applicatif *Stockage*,
- l'attribut *Taille* de la donnée physique *Taille de stockage de la messagerie* est produit par le composant applicatif *Stockage*,
- l'attribut *Date* de la donnée physique *Réception du message* est produit par le composant applicatif *Messagerie*,
- l'attribut *Indicateur* de la donnée physique *Filtrage de message* est produit par le composant applicatif *Messagerie*,
- l'attribut *Libellé* de la donnée physique *Règle de filtrage objet* est produit par le composant applicatif *Filtrage*,
- les relations de dépendance
 - o de la donnée physique *Date de stockage* vers la donnée physique *Réception du message*,
 - o de la donnée physique *Date de stockage* vers la donnée physique *Filtrage du message*,
 - o de la donnée physique *Taille de stockage de la messagerie* vers la donnée physique *Réception du message*,
 - o de la donnée physique *Taille de stockage de la messagerie* vers la donnée physique *Filtrage du message*,sont conformes à l'utilisation de l'interface applicative *Filtrer un message reçu* par le composant applicatif *Stockage* (relations entourées et reliées dans l'illustration),
- les relations de dépendance
 - o de la donnée physique *Réception du message* vers la donnée physique *Règle de filtrage objet*,
 - o de la donnée physique *Filtrage du message* vers la donnée physique *Règle de filtrage objet*,sont conformes à l'utilisation de l'interface applicative *Consulter les règles* par le composant applicatif *Messagerie* (relations entourées et reliées dans l'illustration).

4.2.5.4 Architecture applicative détaillée

L'activité d'architecture applicative détaillée, entourée dans la **Figure 111**, permet de détailler :

- les composants applicatifs conçus lors de l'activité d'architecture applicative statique (cf. §4.2.5.1),
- les interfaces applicatives conçus lors de l'activité d'architecture applicative dynamique (cf. §4.2.5.2) et
- les données physiques conçus lors de l'activité d'architecture applicative dynamique des données (cf. §4.2.5.3).

La contrainte est exprimée sous forme d'un modèle de couches applicatives, par exemple la couche de présentation, la couche d'accès aux données, etc. Chaque composant applicatif et ses interfaces sont projetés sur chacune des couches applicatives. Un environnement architectural, spécifique à chaque couche applicative et conçu lors de l'activité d'architecture technique détaillée (cf. §4.2.4.3), permet d'automatiser l'activité d'architecture applicative détaillée. Cette automatisation à partir d'un environnement architectural associé à une couche applicative est une application classique de l'approche MDA.

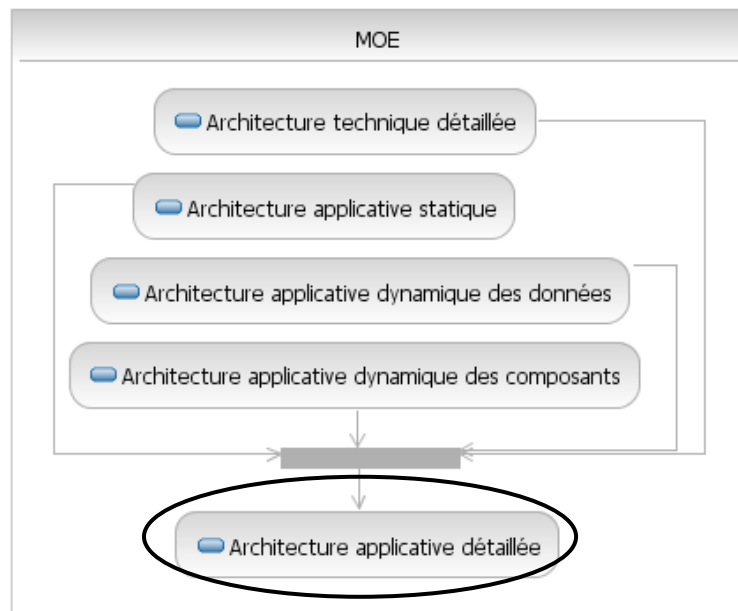


Figure 111 : Activité d'architecture applicative détaillée

L'activité d'architecture applicative détaillée n'est pas explicitée ici car elle ne comporte pas de spécificité dans la démarche EA4UP.

4.2.6 Implémentation

L'implémentation du système est une macro-activité sous la responsabilité de l'équipe de développement, soit la MOE du système. Cette macro-activité, entourée dans la **Figure 112**, nécessite que le système possède une architecture applicative détaillée. L'objectif de l'implémentation est l'écriture du code dans chaque élément constituant les différentes couches applicatives du système.

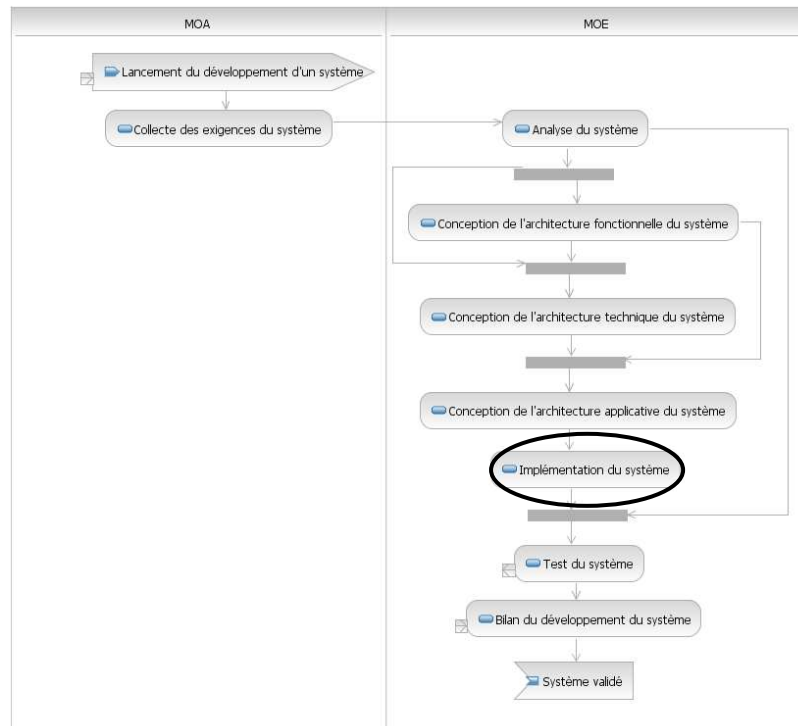


Figure 112 : Macro-activité d'implémentation dans la démarche EA4UP

L'activité d'implémentation entourée dans la **Figure 113** permet de coder les éléments résultants de la projection de chaque composant applicatif, de chaque interface applicative ou de chaque donnée physique sur une couche applicative. Ces éléments issus des différentes projections sont conçus lors de l'activité d'architecture applicative détaillée (cf. §4.2.5.4). Un générateur de code spécifique à un constituant d'une couche applicative permet en général d'automatiser l'activité d'implémentation. La génération de code ainsi automatisée est une application classique de l'approche MDA.

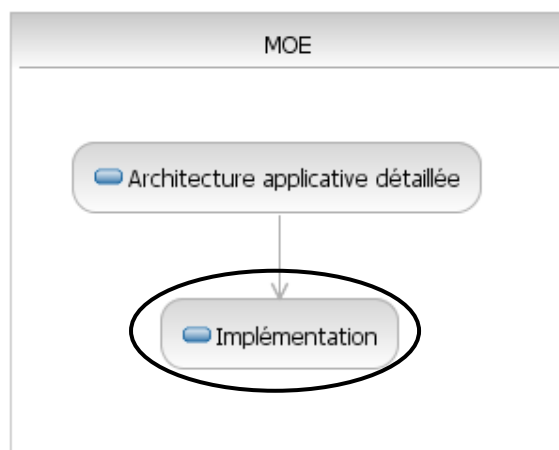


Figure 113 : Activité d'implémentation

L'activité d'implémentation n'est pas explicitée ici car elle ne comporte pas de spécificité dans la démarche EA4UP.

4.2.7 Test

Le test du système est une macro-activité sous la responsabilité conjointe du client du système, soit la MOA, et de l'équipe de développement, soit la MOE du système. La MOE doit tester le système qu'elle développe alors que la MOA doit effectuer la recette du système à partir des résultats des tests. Cette macro-activité, entourée dans la **Figure 114**, nécessite que le système soit implémenté. La macro-activité de test comprend plusieurs activités telles que le test fonctionnel et le test de robustesse et de performance.

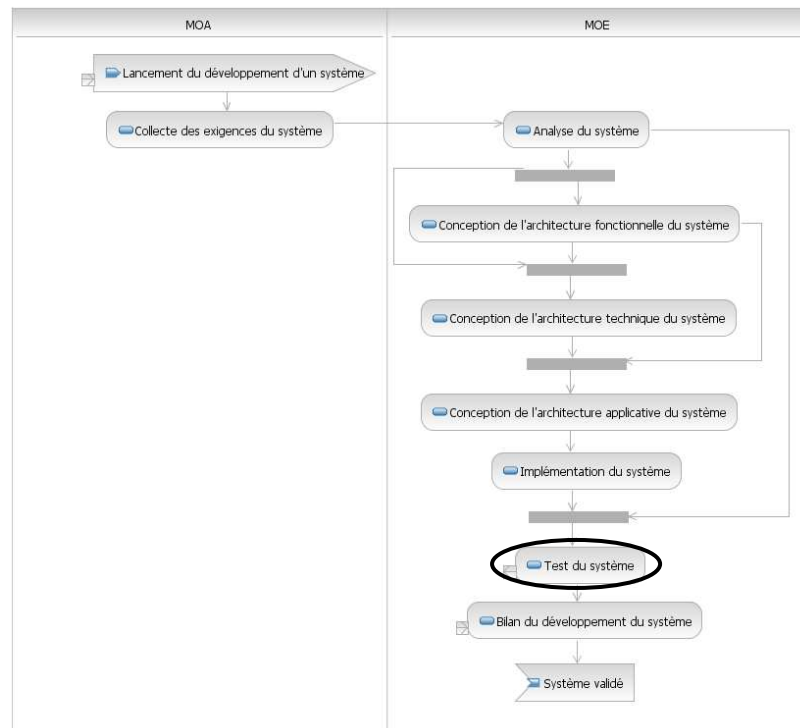


Figure 114 : Macro-activité de test dans la démarche EA4UP

L'activité de test fonctionnel, entourée dans la **Figure 115**, permet, à partir de la reformulation des exigences fonctionnelles réalisée lors de l'activité d'analyse fonctionnelle dynamique (cf. §4.2.2.2), de tester les éléments codés lors de l'activité d'implémentation (cf. §4.2.6). Ces éléments constituent des composants applicatifs, des interfaces applicatives ou des données physiques.

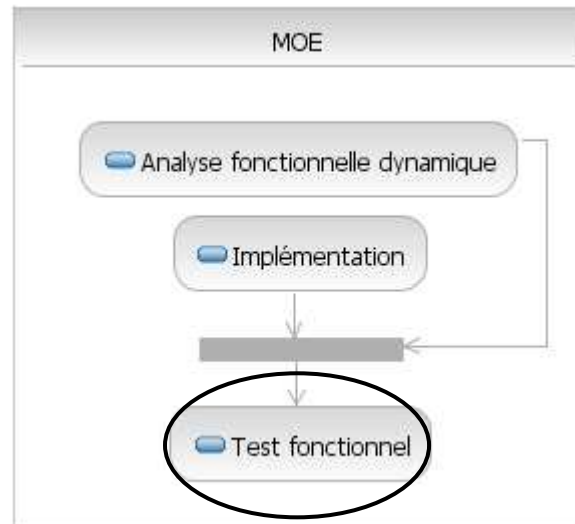


Figure 115 : Activité de test fonctionnel

L'activité de test de robustesse et de performance, entourée dans la **Figure 116**, permet à partir de la reformulation des exigences non fonctionnelles réalisée lors de l'activité d'analyse technique dynamique (cf. §4.2.2.4), de tester les éléments codés lors de l'activité d'implémentation (cf. §4.2.6). Ces éléments constituent les composants applicatifs, les interfaces applicatives ou les données physiques.

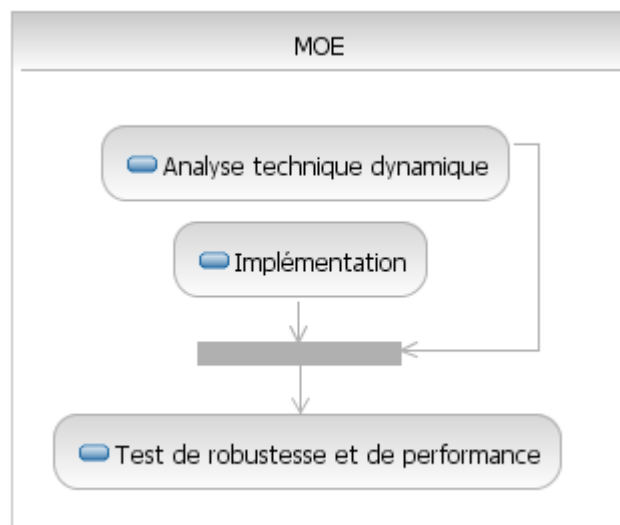


Figure 116 : Activité de test de robustesse et de performance

4.3 Évaluation de la démarche EA4UP

L'activité de réalisation du bilan du développement d'un système, entourée dans la **Figure 117**, recouvre, lors du prototypage de la démarche EA4UP, son évaluation. La démarche a été prototypée par plusieurs projets de développement de services télécom.

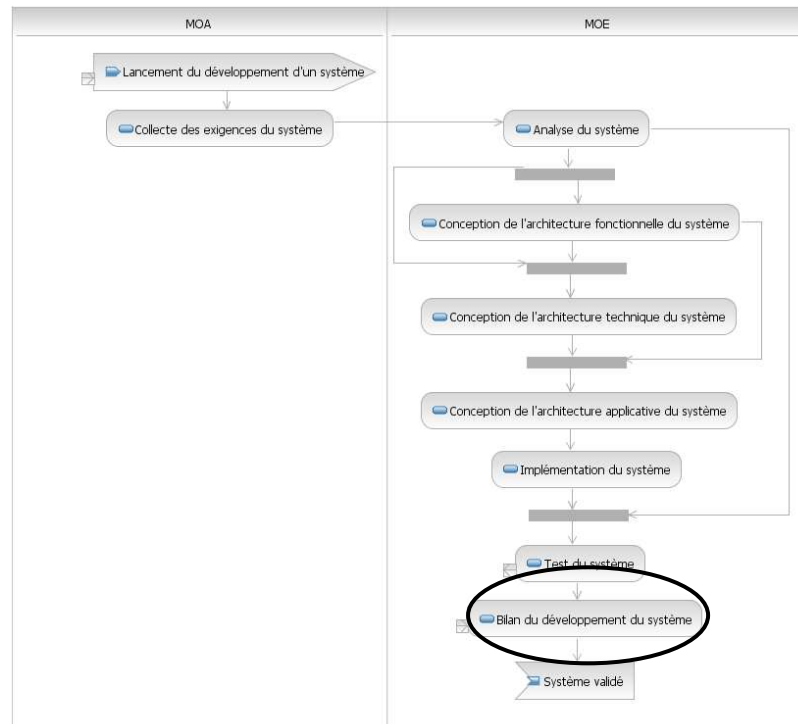


Figure 117 : Activité liée au bilan du développement dans la démarche DA4EA

Durant le premier semestre 2007, la démarche EA4UP a été prototypée par onze projets (6 projets de Telekomunikacja Polska R&D et 5 projets de France Télécom R&D). Parmi ces onze projets, neuf étaient des projets de développement de service télécom. Le processus de développement durant le prototypage était limité aux macro-activités de collecte des exigences, d'analyse et de conception de l'architecture, à l'exception des activités d'architecture technique détaillée et d'architecture applicative détaillée. Les deux projets de développement restant ne ciblaient que les macro-activités de collecte des exigences, d'analyse et de conception de l'architecture fonctionnelle. Cependant, pour l'un des deux, des architectures applicatives de système ont été cartographiées a posteriori sur l'architecture fonctionnelle conçue avec la démarche EA4UP.

4.3.1 Contexte du prototypage de la démarche EA4UP

Le prototypage de la démarche EA4UP est constitué d'un support du développement du système par le concepteur de la démarche, puis d'un questionnaire à partir duquel le bilan du prototypage a été réalisé.

Le questionnaire contient dix questions sur les thèmes suivants :

- l'approche EA4UP, avec une évaluation de la complexité de l'approche par rapport à son intérêt pour l'architecte du système,
- le contenu architectural issu de l'application de la démarche EA4UP, avec une évaluation de l'architecture obtenue et une estimation de la qualité de la conception, en particulier en termes de traçabilité,
- le coût de l'approche EA4UP comparé au coût de l'approche de développement utilisée précédemment,
- l'apport de la démarche EA4UP dans le domaine de la communication vers les architectes d'autres projets, vers le marketing et vers les décideurs.

Le questionnaire envoyé aux projets participant au prototypage est reporté ci-après dans sa version originale.

4.3.1.1 Approche EA4UP

Q1. Do you find this approach "very complicated — complicated — simple — very simple" for using?

Q2. Do you find this approach "without interest — not very useful — useful — essential" for your architect work?

Q3. Is the approach support (1 day for architecture designing deducing from a use case + architecture document second reading) "without interest — not very useful — useful — essential" for the approach using?

4.3.1.2 Architecture et approche EA4UP

Q4. Is with this approach the architecture component reusability "definitely worse — worse — better — definitely better" than the component reusability of your previous architecture approach?

Q5. Is the quality of the architecture designed with this approach "definitely worse — worse — better — definitely better" than the quality of your previous architecture approach?

4.3.1.3 Coût de l'approche EA4UP

Q6. Is the architecture designing time "definitely longer — longer — shorter — definitely shorter" than the previous architecture designing time?

4.3.1.4 Communication et approche EA4UP

Q7. To communicate the results of the architecture design, do you think that this approach "does not help — helps a little — give sounds background — is essential"?

Q8. To work in common with marketing team, have you find this approach "without interest — not very useful — useful — essential"?

Q9. To work in common with architects, have you find this approach "without interest — not very useful — useful — essential"?

Q10. To interact regularly with decision makers have you find this approach "without interest — not very useful — useful — essential"?

4.3.2 Bilan du prototypage de la démarche EA4UP

Les réponses de chaque projet (P1 à P11) sont reportées dans le Tableau 15 de la façon suivante :

- Q1 : approach very complicated "—", complicated "-", simple "+", very simple "++",
- Q2 : approach without interest "—", not very useful "-", useful "+", essential "++",
- Q3 : approach without interest "—", not very useful "-", useful "+", essential "++",
- Q4 : component reusability definitely worse "—", worse "-", better "+", definitely better "++",
- Q5: quality definitely worse "—", worse "-", better "+", definitely better "++"),
- Q6 : designing time definitely longer "—", longer "-", shorter "+", definitely shorter "++",
- Q7 : approach does not help "—", helps a little "-", give sounds background "+", is essential "++",
- Q8 : to work with marketing team approach is without interest "—", not very useful "-", useful "+", essential "++",
- Q9 : to work with architects approach is without interest "—", not very useful "-", useful "+", essential "++",
- Q10 : to work with decision makers approach is without interest "—", not very useful "-", useful "+", essential "++".

Projet	Approche			Architecture		Coût	Communication				Compétence EA
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	
P1	-	++	+	++	+	+	--		++	++	N
P2	-	++	++	+		--			++		N
P3	+	++	++	++	+	+	+	-	+	-	O
P4	-	+	+	+	+	-	+	-	+	+	N
P5	-	+	++				+	-	-	--	N
P6	+	++	+	+	++	++	++	++	++	++	O
P7	-	+	++	++	++	-	++	+	+	+	N
P8	+	+	++	++	+	-	+		+	+	N
P9	-	+	++				+		+	+	N
P10	+	++		+	+	++	++	+	++	+	O
P11	-	++	++	++	++	-	+	+	++		N

Tableau 15 : Réponses au questionnaire du bilan de la démarche EA4UP.

La dernière colonne indique la compétence acquise au préalable par l'architecte sur l'EA fonctionnelle du SI des services télécom.

4.3.3 Analyse du bilan du prototypage de la démarche EA4UP

Le bilan de l'approche EA4UP décrit dans la **Figure 118** indique de bons résultats pour la réutilisabilité des composants applicatifs, symbole de la durabilité d'un SI. Un moins bon résultat est celui de la comparaison du coût de développement de la démarche EA4UP par rapport aux démarches utilisées précédemment. La différence de coût est due à l'appropriation de la vue fonctionnelle de l'EA du SI par chaque architecte. Une compétence en EA fonctionnelle de l'architecte est en effet corrélée avec un coût moindre de la démarche EA4UP.

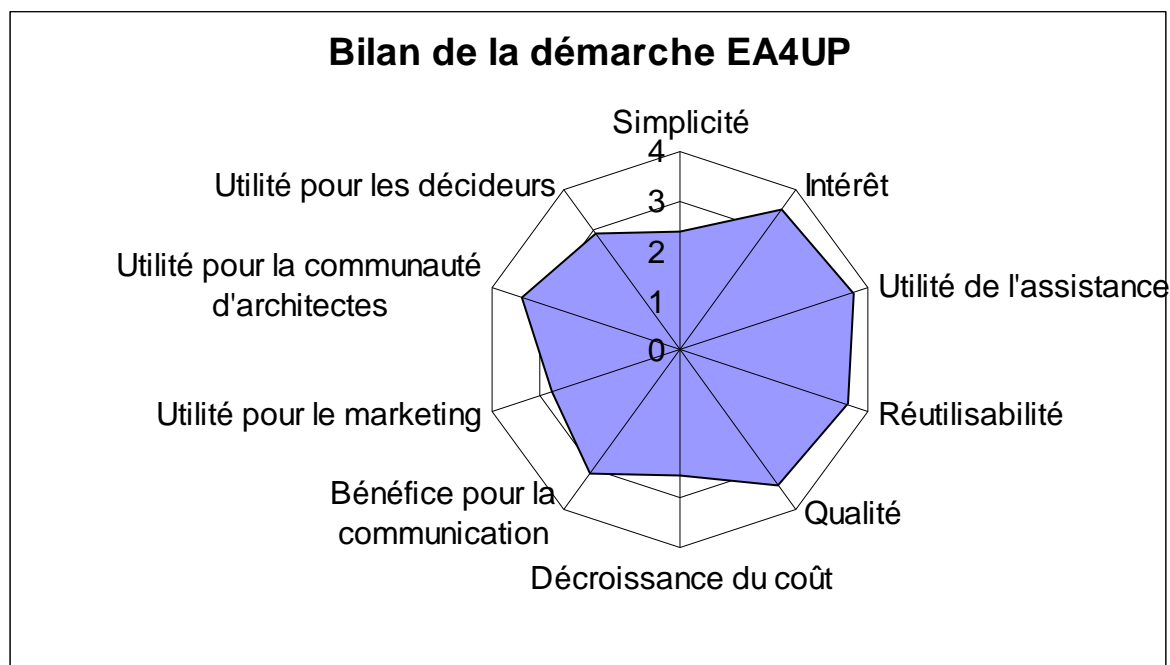


Figure 118 : Bilan de la démarche EA4UP

En dépit de sa complexité, la démarche EA4UP a une bonne évaluation de son intérêt par le projet. L'utilité de l'assistance fournie par le concepteur de la démarche lors du prototypage est néanmoins soulignée par la plupart des projets.

Pour tous les projets, la démarche EA4UP permet d'améliorer la pertinence de l'architecture de leur service et la qualité de cette architecture. Cette qualité est particulièrement soulignée par la traçabilité de chaque élément conçu pour l'architecture du service télécom par rapport à l'analyse fonctionnelle de ce service.

Enfin, l'amélioration de la communication est significative avec les architectes d'autres projets de services télécom. Cette amélioration favorise une meilleure réutilisabilité des composants applicatifs conçus par le projet.

Elle reste néanmoins à prouver pour la communication avec le marketing, et à un degré moindre pour la communication avec les décideurs.

4.4 Conclusion

4.4.1 Démarche EA4UP et SOA

La démarche EA4UP (cf. §4.2) s'inscrit dans la notion de durabilité d'un SI telle que définie dans le §2.1.1. La réutilisation des différents éléments conçus et codés dans un SI est une caractéristique importante de la durabilité du SI. L'architecture SOA est une architecture qui peut bénéficier d'une telle propriété. La difficulté est de mettre en œuvre un processus de développement permettant de faire bénéficier l'architecture SOA de la réutilisation au niveau d'un SI telle que ciblée par le cadre de l'urbanisme. L'approche EA4UP décrite dans le §0 permet de définir un tel processus de développement. La notion de service associée à l'architecture SOA peut en effet être définie dans l'approche EA4UP par l'accès dynamique à une donnée. La vue métier des processus d'une entreprise est remplacée dans notre approche par la vue fonctionnelle d'un système du SI de l'entreprise. Le lien entre la vue fonctionnelle et la vue métier est néanmoins vérifié lors de l'activité d'alignement de la démarche DA4UP (cf. §3.4). Cette activité permet en effet de mesurer l'alignement de la vue fonctionnelle du SI, qui contraint la vue fonctionnelle du système, avec la vue métier de l'entreprise (cf. §3.4).

Lors de l'activité d'architecture fonctionnelle dynamique des données de la démarche EA4UP (cf. §4.2.3.3), une séquence d'interactions entre îlots fonctionnels représente une architecture de type SOA. Le diagramme de séquence représentant l'architecture fonctionnelle dynamique peut être considéré comme une orchestration de services enrichie de dépendances entre services. Cet enrichissement est induit par l'approche dynamique de la démarche EA4UP.

Par exemple, le diagramme de séquence de la **Figure 90** correspond à l'architecture fonctionnelle SOA représentée dans la **Figure 119**. Chaque service fonctionnel est représenté graphiquement par une interface de composant UML.

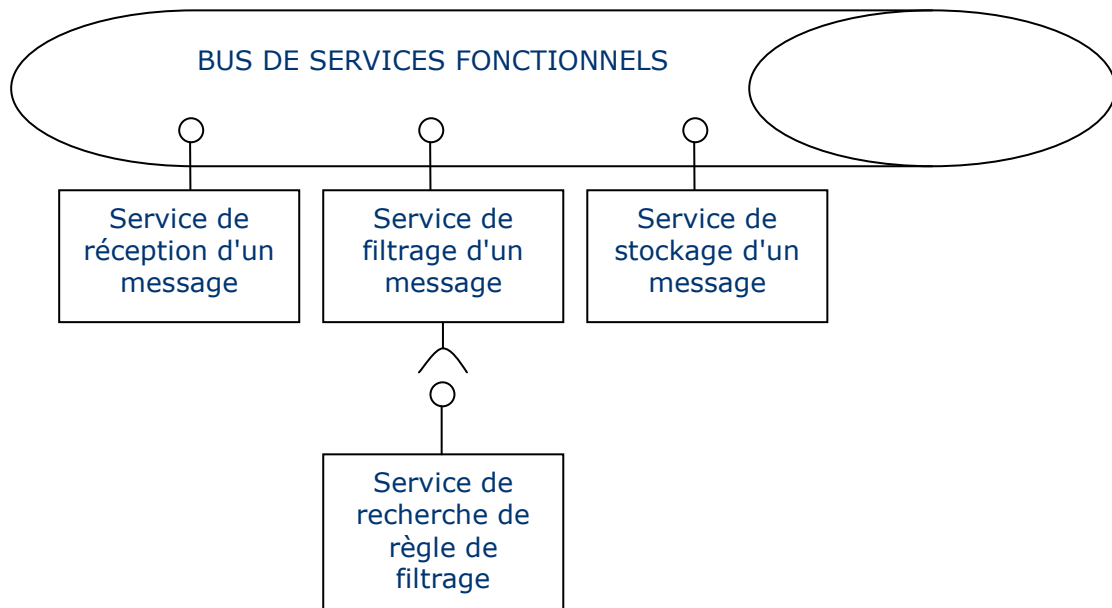


Figure 119 : Illustration d'une orchestration de services fonctionnels

Une conséquence importante du point de vue technique est de faciliter ainsi l'optimisation de l'utilisation du bus de services. Les appels aux services fonctionnels du point de vue du débit de données logiques sont en effet optimisés par l'architecture de l'orchestration déduite de la démarche EA4UP.

D'autres scénarios de cas d'utilisation peuvent cependant nécessiter de rendre accessible, directement à partir du bus de services, les services dont dépendent les services offerts par cette architecture SOA. Les services offerts sont, du fait de la démarche, conformes au PLU fonctionnel, et donc à l'urbanisme fonctionnel du SI.

L'architecture d'une orchestration de services de la vue applicative est conforme à la fois à l'urbanisme fonctionnel du SI, mais aussi à l'urbanisme technique de ce SI. Le diagramme de séquence de la **Figure 110** représente l'architecture applicative SOA décrite dans la **Figure 120**.

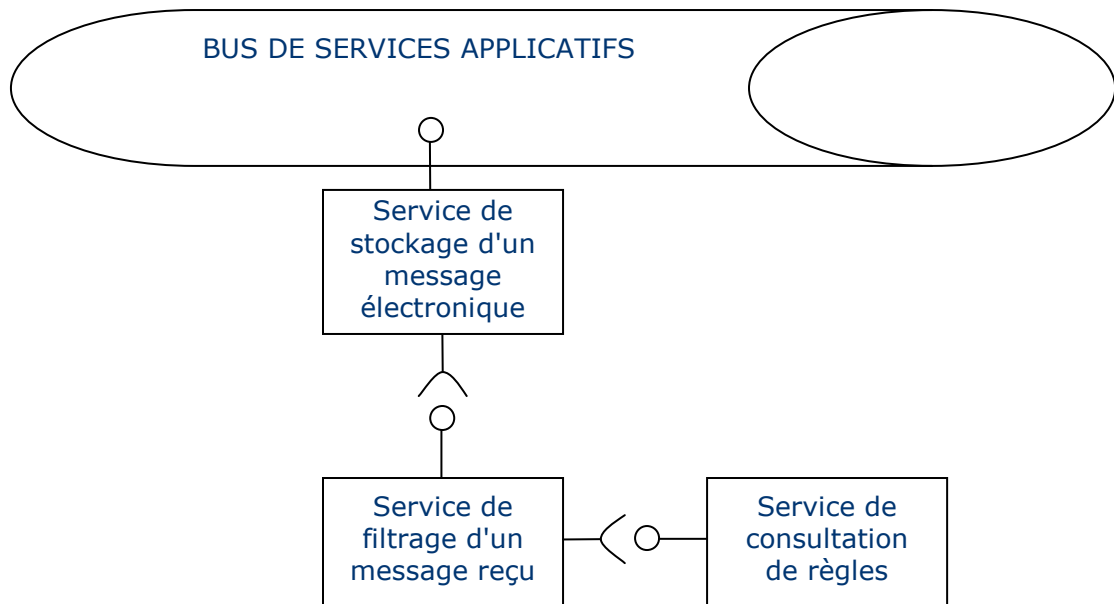


Figure 120 : Illustration d'une orchestration de services applicatifs

L'optimisation de l'utilisation du bus de services applicatifs est caractérisée lors de l'accès au bus de services par un seul accès au service de stockage d'un message électronique. La difficulté du dimensionnement du bus de services est ainsi redistribuée sur les liens de dépendance entre services applicatifs. Le dimensionnement de ces liens est immédiat puisque ne caractérise qu'une seule relation de dépendance entre services applicatifs.

La démarche EA4UP et son approche dynamique permet aussi de s'assurer du couplage faible entre les services accédés directement à partir du bus, qualité nécessaire d'une architecture SOA. Les différents livrables de l'architecture fonctionnelle dynamique ou de l'architecture applicative dynamique permettent de représenter le couplage entre les appels de service successifs réalisant un scénario d'usage d'un système. Les interactions entre îlots fonctionnels fortement dépendants ou entre composants applicatifs fortement dépendants sont en effet représentées dans ces livrables de l'approche dynamique.

C'est le cas de la dépendance fonctionnelle du filtrage du message vis-à-vis de la recherche de la règle de filtrage pour les services fonctionnels de la **Figure 119** et des dépendances applicatives du stockage d'un message vis-à-vis du filtrage d'un message reçu et du filtrage d'un message reçu vis-à-vis de la consultation de règles pour les services applicatifs de la **Figure 120**.

4.4.2 Bilan de la démarche EA4UP

La synthèse de la démarche EA4UP (cf. §4.3) faite sur les quatre thèmes proposés pour le bilan met en évidence dans la **Figure 121** :

- la complexité de l'approche EA4UP,
- l'apport de la démarche EA4UP lors de la conception de l'architecture d'un service télécom,
- l'augmentation du coût de développement lié à l'utilisation de la démarche EA4UP,
- l'apport inégal pour la communication avec des tiers.

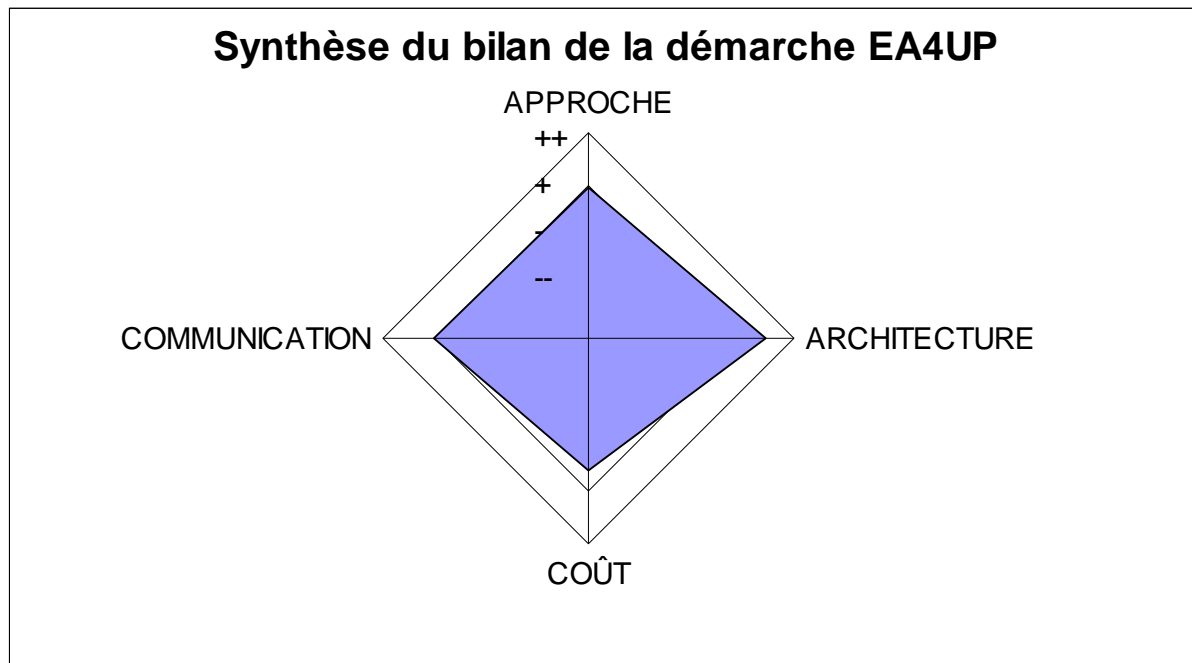


Figure 121 : Synthèse du bilan de la démarche EA4UP

Le très bon résultat de la conception de l'architecture au vu de la réutilisabilité des éléments conçus et de la qualité de la conception en termes de traçabilité souligne l'aspect durable d'un SI dont les systèmes sont conçus avec la démarche EA4UP. Le mauvais résultat du coût et de la simplicité de l'approche nécessite une aide outillée telle que celle proposée dans la démarche DA4EA. Pour la communication, cette aide outillée devra prendre en compte le mode de représentation le plus cohérent avec la population visée (marketing ou décideur).

5. Démarche EA4UP et mesure d'alignement fonctionnel de type information mutuelle moyenne

5.1 Principes

En reprenant le processus d'usage d'un logement (cf. §2.3.2), l'objectif de la mesure d'alignement fonctionnel associée à la démarche EA4UP est d'évaluer l'écart fonctionnel entre :

- un logement en cours de construction avec la démarche EA4UP et,
- le logement modèle conçu avec la démarche DA4EA.

Cet écart fonctionnel représente le non respect par l'architecte du logement en cours de construction du modèle fonctionnel de référence pour un logement.

Un alignement associé à la démarche de développement de système EA4UP nécessite une définition du point de vue fonctionnel, du point de vue technique et du point de vue applicatif entre le système et la cible définie par les urbanistes. Dans les diagrammes d'activités décrivant la démarche EA4UP, les activités de développement de système interagissent en effet avec des événements émis :

- par les urbanistes fonctionnels, pour l'évènement de validation du PLU fonctionnel,
- par les urbanistes techniques, pour l'évènement de validation des préconisations techniques,
- par les urbanistes applicatifs, pour l'évènement de validation d'éléments applicatifs réutilisables.

L'alignement avec l'EA lors du développement d'un système est étudié dans ce chapitre entre la vue fonctionnelle du système et la vue fonctionnelle de l'EA du SI auquel appartient le système.

5.1.1 Définition de l'alignement fonctionnel

La conception de la vue fonctionnelle d'un système interagit avec l'évènement de validation du PLU fonctionnel lors des activités d'architecture fonctionnelle statique (cf. §4.2.3.1) et d'architecture fonctionnelle dynamique des îlots (cf. §4.2.3.2). L'approche dynamique est, comme pour les démarches DA4EA et EA4UP, privilégiée pour la définition de l'alignement de la vue fonctionnelle d'un système avec l'EA fonctionnelle du SI. L'alignement prend ainsi en compte la dynamique entre les îlots fonctionnels réalisant les scénarios d'utilisation du système.

Le premier échange entre le développement d'un système et les résultats de l'EA se situe lors du codage par un expert des éléments d'analyse fonctionnelle statique par les parcelles fonctionnelles issues d'îlots du PLU fonctionnel (cf. §4.2.3.1). Les impacts du codage sont significatifs lors de l'activité d'architecture fonctionnelle dynamique des îlots (cf. §4.2.3.2). L'automatisation de l'alignement des interactions entre entités avec les interactions entre îlots fonctionnels est en effet une conséquence de ce codage.

Le second échange entre le développement d'un système et l'EA fonctionnelle se situe lors de l'activité d'architecture fonctionnelle dynamique des îlots. Il s'agit des voies du PLU fonctionnel qui permettent à certains îlots fonctionnels d'interagir.

Afin de bénéficier de la richesse du PLU fonctionnel, le concept de voie fonctionnelle entre îlots est étendu au niveau quartier et au niveau zone par le concept de voie du PLU fonctionnel.

Définition : une voie du PLU fonctionnel est définie

- entre deux quartiers fonctionnels, s'il existe une voie fonctionnelle entre un îlot fonctionnel contenu dans un quartier vers un îlot fonctionnel appartenant à l'autre quartier. L'orientation de la voie entre deux quartiers est cohérente avec l'orientation de toutes les voies fonctionnelles entre les îlots de chaque quartier supposées homogènes pour chaque couple de quartiers (cf. **Figure 122**);

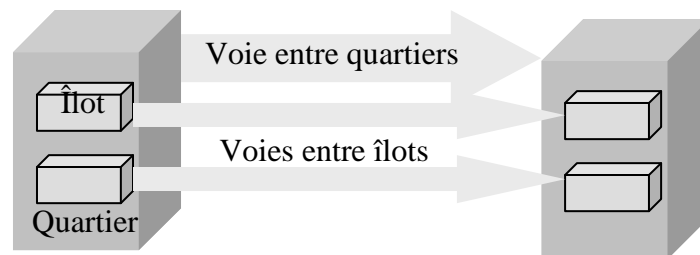


Figure 122 : Voie du PLU entre quartiers fonctionnels

- entre deux zones fonctionnelles s'il existe une voie fonctionnelle entre un îlot fonctionnel contenu dans une zone vers un îlot fonctionnel appartenant à l'autre zone. L'orientation de la voie entre deux zones est cohérente avec l'orientation des voies fonctionnelles entre îlots de chaque zone supposées homogènes pour chaque couple de zones (cf. **Figure 123**).

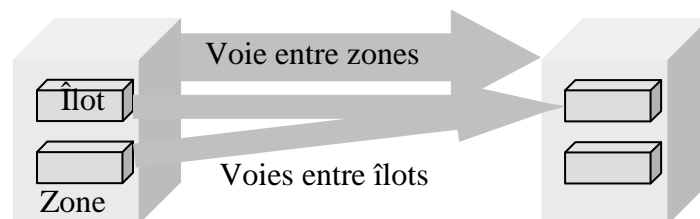


Figure 123 : Voie du PLU entre zones fonctionnelles

Dans ce cas, les concepts à aligner sont pour la vue fonctionnelle du système :

- l'entité spécifiée lors de l'analyse fonctionnelle statique (cf. §4.2.2.1),
- l'interaction entre entités spécifiée lors de l'analyse fonctionnelle dynamique (cf. §4.2.2.2),

et pour l'EA du SI contenant le système :

- l'îlot fonctionnel conçu lors de l'architecture fonctionnelle statique (cf. §4.2.3.1),
- l'interaction entre îlots fonctionnels conçue lors de l'architecture fonctionnelle dynamique des îlots (cf. §4.2.3.2),
- la voie du PLU fonctionnel déduite d'une voie fonctionnelle conçue de façon automatisée dans la démarche DA4EA (cf. §3.6.1.3).

L'alignement de la vue fonctionnelle d'un système avec l'EA fonctionnelle est estimé lors de l'activité d'architecture fonctionnelle dynamique, c'est-à-dire lors de la conception des interactions réalisant un scénario d'utilisation d'un système.

Définition : L'alignement fonctionnel est défini entre les interactions entre entités et les interactions entre îlots fonctionnels. L'alignement est d'autant meilleur que les interactions entre îlots fonctionnels respectent la Règle d'urbanisme – Interactions et voies (cf. §3.2) pour des voies du PLU fonctionnel.

La Règle d'urbanisme – Alignement fonctionnel définit la conformité des interactions entre îlots fonctionnels alignées avec des interactions entre entités et les voies du PLU fonctionnel.

Règle d'urbanisme – Alignement fonctionnel n°17

Une interaction entre deux instances d'îlots fonctionnels alignée avec une interaction entre instances d'entités est conforme avec une voie du PLU fonctionnel si :

- lorsque les deux îlots fonctionnels codent une même entité qui est la source ou la cible d'une interaction, alors il existe une voie du PLU fonctionnel entre les deux îlots fonctionnels ou entre les deux quartiers les contenant ou entre les deux zones les contenant,
- lorsqu'il existe une interaction entre entités de type requête, respectivement de type réponse, alignée avec une interaction de type requête, respectivement de type réponse, entre îlots fonctionnels et que cette interaction entre îlots respecte la Règle d'urbanisme – Interactions et voies avec les voies du PLU fonctionnel.

Dans l'illustration de la **Figure 124**, l'alignement fonctionnel est estimé pour le scénario instanciant le cas d'utilisation *CU*.

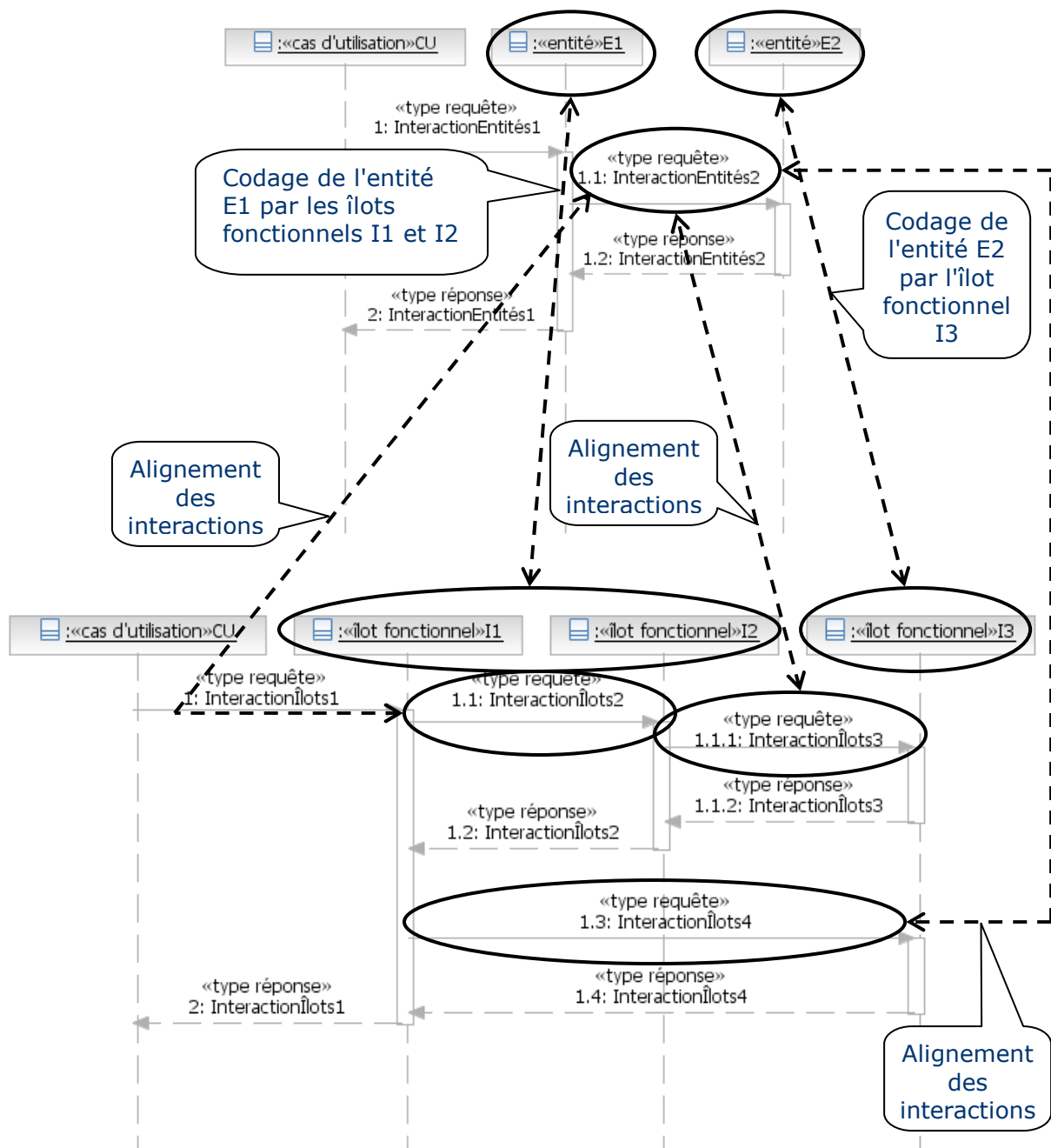


Figure 124 : Alignement fonctionnel lors de l'activité d'architecture fonctionnelle dynamique des îlots

Si :

- l'entité E1 est codée par les îlots fonctionnels I1 et I2,
- l'entité E2 est codée par l'îlot fonctionnel I3,
- l'interaction entre entités *InteractionEntités2* est alignée avec les interactions entre îlots fonctionnels *InteractionÎlots2*, *InteractionÎlots3* et *InteractionÎlots4* d'après la définition de l'alignement d'interactions du §4.2.3.2,

alors l'alignement fonctionnel du système sur l'EA fonctionnelle dépend de l'existence dans le PLU fonctionnel :

- d'une voie entre l'îlot fonctionnel I1 et l'îlot fonctionnel I2, ou d'une voie du PLU fonctionnel entre les quartiers fonctionnels contenant I1 et I2 ou entre les zones fonctionnelles contenant I1 et I2,
- d'une voie de l'îlot fonctionnel I2 vers l'îlot fonctionnel I3, ou d'une voie du PLU fonctionnel entre les quartiers fonctionnels contenant I2 et I3 ou entre les zones fonctionnelles contenant I2 et I3,
- d'une voie de l'îlot fonctionnel I1 vers l'îlot fonctionnel I3, ou d'une voie du PLU fonctionnel entre les quartiers fonctionnels contenant I1 et I3 ou entre les zones fonctionnelles contenant I1 et I3.

5.1.2 Axiomatisation de l'alignement fonctionnel

Le comportement attendu de l'alignement fonctionnel défini dans le §5.1.1 est spécifié dans les deux axiomes suivants. Les axiomes de l'Alignement Fonctionnel d'un Système sur l'EA (AFS) du SI auquel appartient le système ciblent les concepts d'analyse fonctionnelle du système.

AFS1 – L'alignement fonctionnel doit être dépendant de l'analyse fonctionnelle et de l'architecture fonctionnelle du système. Par contre, cet alignement doit être indépendant de la granularité des scénarios spécifiés lors de la collecte des exigences fonctionnelles dynamique (cf. §4.2.1.2).

La granularité de chaque scénario est en effet liée aux conditions de la collecte des exigences. Aucune règle n'existe sur la granularité des cas d'utilisation d'un système et des scénarios qui en sont dérivés. La découpe des fonctionnalités du système offertes à l'utilisateur est de ce fait très aléatoire. Dans l'illustration de la réception d'un message et de son filtrage, doit-on par exemple séparer le cas d'utilisation *CU-Réception de message avec filtrage* en deux cas d'utilisation : *CU-Réception de message* et *CU-Filtrage de message* ? Une réponse homogène à ce type d'interrogation durant la collecte des exigences fonctionnelles est difficile à réguler, voire même à obtenir.

AFS2 – L'alignement fonctionnel doit prendre en compte la richesse du PLU fonctionnel grâce à l'extension du concept de voie fonctionnelle au niveau quartier et au niveau zone du PLU fonctionnel.

Du fait de la définition d'une voie du PLU fonctionnel, l'alignement d'une interaction entre entités avec une interaction entre îlots fonctionnels cohérente avec une voie fonctionnelle entre îlots est meilleur que si l'interaction entre îlots est cohérente avec une voie du PLU fonctionnel au niveau quartier ou au niveau zone. De même, l'alignement est meilleur si l'interaction entre îlots est cohérente avec une voie du PLU fonctionnel au niveau quartier que si l'interaction entre îlots est cohérente avec une voie du PLU fonctionnel au niveau zone.

5.1.3 Choix d'une mesure de l'alignement fonctionnel

La mesure de l'alignement fonctionnel axiomatisée dans le §5.1.2 doit permettre une optimisation de la conception de l'architecture fonctionnelle du système [Woodside, 2007]. La mesure de l'alignement fonctionnel décrite dans [Simonin, 2007] ne peut être choisie du fait de l'axiome AFS1. En effet, cette mesure dans le cas dynamique est paramétrée par la longueur moyenne des séquences d'interactions cohérentes avec des voies du PLU fonctionnel. La longueur des séquences dépendant de la granularité des scénarios collectés, l'axiome AFS1 n'est pas vérifié.

La mesure choisie est l'information mutuelle moyenne [Shannon, 1948]. L'information définie par Shannon est actuellement un concept plus abstrait [Wang, 2002] qu'à l'origine de la théorie de l'information. Pour l'alignement fonctionnel évalué ici, l'émission et la réception caractérisent une interaction entre entités. L'idée est d'associer le système de communication lié à cette interaction à l'alignement fonctionnel défini précédemment (cf. §5.1.1). La qualité de la transmission est alors paramétrée par la nature de la voie du PLU fonctionnel cohérente avec chaque interaction entre îlots fonctionnels. La perte d'information lors de la transmission est minimum lorsque la voie est une voie fonctionnelle entre îlots et elle est maximum lorsqu'il n'y a pas de voie du PLU fonctionnel entre les zones contenant les îlots de l'interaction.

L'émission, la réception et le système de communication tels que définis en théorie de l'information sont représentés dans la **Figure 125** avec l'exemple de l'interaction entre entités *InteractionEntités2*.

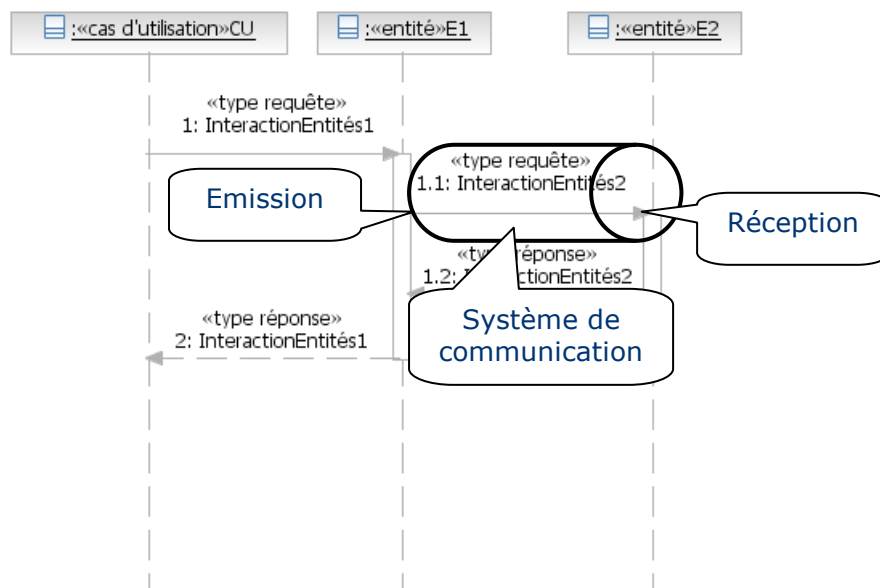


Figure 125 : Alignement fonctionnel et théorie de l'information

Définition : Soit X , la variable aléatoire représentant l'émission d'une interaction de l'ensemble Int des interactions entre entités réalisant des scénarios d'usage d'un système, et Y , la variable aléatoire représentant la cible d'une interaction de Int , alors la mesure MAFS de l'alignement de la vue fonctionnelle d'un système par rapport à l'EA fonctionnelle du SI auquel appartient le système est telle que :

$$MAFS(Int) = I(X(Int); Y(Int))$$

Équation 5-1

avec $I(X(Int); Y(Int))$, l'information mutuelle moyenne apportée par la connaissance de la cible d'une interaction sur l'émission de l'interaction.

La mesure MAFS prend ses valeurs entre 0, lorsque les évènements réalisés par X et par Y sont indépendants, et l'entropie de la variable aléatoire Y si les évènements sont entièrement dépendants. En effet, si H est l'entropie, par définition de l'information mutuelle moyenne, $I(X(Int); Y(Int)) = H(Y(Int)) - H(Y(Int)|X(Int))$. La relation entre l'information mutuelle moyenne et l'entropie est représentée dans le diagramme de Venn de la **Figure 126**.

Si les évènements réalisés par X et par Y sont indépendants, alors $H(Y|X) = H(Y) \Rightarrow I(X(Int); Y(Int)) = 0$. Si les évènements sont dépendants, alors $H(Y|X) = 0 \Rightarrow I(X(Int); Y(Int)) = H(Y)$.

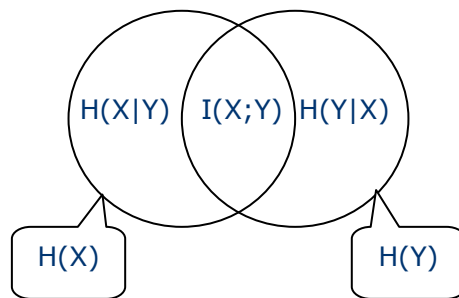


Figure 126 : Information mutuelle moyenne et entropie

Proposition : la mesure MAFS vérifie les deux axiomes AFS1 et AFS2 de l'alignement de la vue fonctionnelle d'un système avec l'EA fonctionnelle du SI auquel appartient le système.

Justification :

- AFS1 : L'information mutuelle moyenne apportée par la connaissance de la cible d'une interaction sur l'émission de l'interaction est évaluée sur chaque interaction réalisant un scénario d'usage d'un service télécom. Le nombre d'interactions successives entre entités réalisant un scénario n'est donc pas pris en compte par la mesure MAFS. La mesure est donc indépendante de la façon dont sont collectées les exigences fonctionnelles du système. La mesure MAFS est dépendante de l'analyse fonctionnelle et de l'architecture fonctionnelle du système puisqu'elle dépend de l'alignement des interactions entre entités spécifiées en analyse et des interactions entre îlots fonctionnels conçus en architecture fonctionnelle.
- AFS2 : Le système de communication étant défini à partir de la nature des voies cohérentes avec chaque interaction entre îlots fonctionnels alignée avec une interaction entre entités, la mesure MAFS dépend du PLU fonctionnel.

5.2 Démarche EA4UP et théorie de l'information

5.2.1 Analyse fonctionnelle dynamique et transmission d'information

L'alignement de la vue fonctionnelle d'un système avec l'EA fonctionnelle du SI est évalué lors de l'activité d'architecture fonctionnelle dynamique des îlots rappelée dans la **Figure 127**. Dans cette même figure, l'activité amont d'analyse fonctionnelle dynamique est étiquetée avec un cercle plein. L'apport de cette activité pour l'alignement fonctionnel est de représenter des interactions entre entités qui donnent lieu à une transmission d'informations.

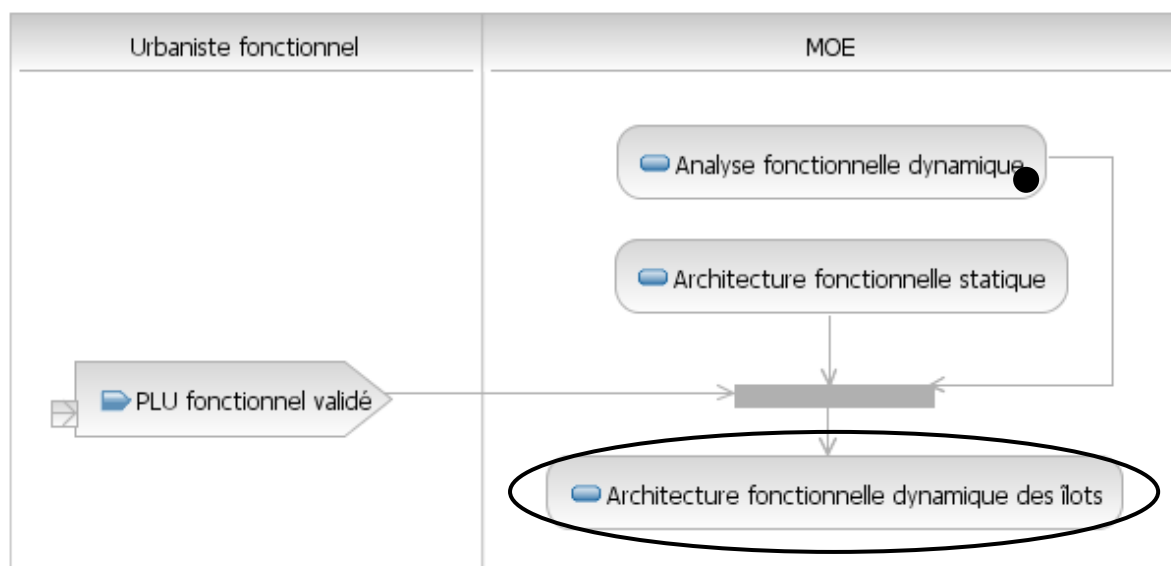


Figure 127 : Analyse fonctionnelle dynamique et activité d'architecture fonctionnelle dynamique des îlots

La transmission d'informations ainsi définie est contrainte par le PLU fonctionnel conçu par les urbanistes fonctionnels.

5.2.2 PLU fonctionnel et contrainte de transmission

La contrainte lors de la transmission d'informations liée à l'analyse fonctionnelle dynamique est représentée par le PLU fonctionnel étiqueté avec un cercle plein dans la **Figure 128**. Du fait de la définition de l'alignement, la conception des îlots et de leurs parcelles, des voies fonctionnelles entre îlots, des quartiers et des zones contenant les îlots contraignent en effet l'architecture dynamique des îlots.

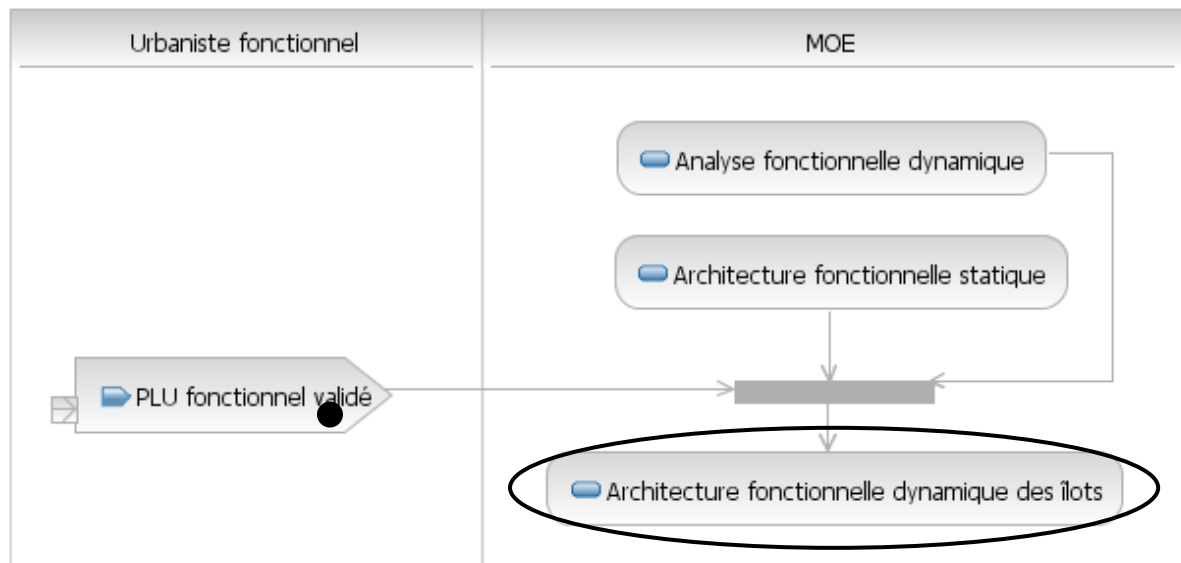


Figure 128 : Contrainte du PLU fonctionnel et activité d'architecture fonctionnelle dynamique des îlots

L'image de la route et du débit en nombre de véhicules à l'heure peut illustrer cette contrainte. Dans le système de communication constitué du PLU fonctionnel, une interaction entre entités est transmise par un système de communication assimilable :

- à une autoroute, lorsque l'interaction entre entités est alignée avec une interaction entre îlots fonctionnels cohérente avec une voie fonctionnelle entre îlots,
- à une voie rapide à quatre voies lorsque l'interaction entre entités est alignée avec une interaction entre îlots fonctionnels cohérente avec une voie du PLU fonctionnel au niveau quartier,
- à une route à deux voies lorsque l'interaction entre entités est alignée avec une interaction entre îlots fonctionnels cohérente avec une voie du PLU fonctionnel au niveau zone.

La route et son débit permettent ainsi de définir un canal de communication entièrement à partir du PLU fonctionnel. Les voies du PLU fonctionnel non limitées aux voies fonctionnelles entre îlots permettent de diminuer notablement les erreurs d'alignement.

5.3 Alignement fonctionnel et système de communication

La transmission d'information est concrétisée par l'existence d'une voie du PLU fonctionnel entre les îlots fonctionnels codant l'entité à l'origine de l'interaction vers les îlots fonctionnels codant l'entité cible de l'interaction. De manière analogue à un système de reconnaissance de la parole [Markowitz, 1996], les erreurs de transmission liées au canal de communication représenté par le PLU fonctionnel sont de deux types :

- élision : il n'y a pas d'îlot fonctionnel accessible par une voie du PLU fonctionnel à partir des îlots codant l'entité à l'origine de l'interaction,
- substitution : les îlots fonctionnels accessibles par une voie du PLU fonctionnel à partir des îlots codant l'entité à l'origine de l'interaction ne codent pas l'entité cible de l'interaction.

La définition du codage et d'un canal de communication assurant plus ou moins parfaitement la transmission via une voie du PLU fonctionnel sont à l'origine de la définition d'un système de communication.

5.3.1 Définition du système de communication

Les concepts d'un système de communication sont représentés dans la **Figure 129**. Ces concepts sont d'abord mis en correspondance avec un concept utile à l'évaluation de l'alignement fonctionnel d'un système avec l'EA fonctionnelle du SI du système.

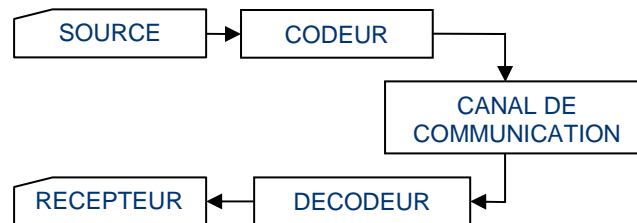


Figure 129 : Système de communication

La source et le récepteur sont caractérisés, respectivement, par l'émission d'une interaction entre entités et par la cible de cette interaction (cf. §5.2.1). De plus, le PLU fonctionnel définit le canal de communication du système de communication d'après le §5.2.2. Le codage de l'émission d'une interaction est alors équivalent au codage de l'entité d'où est émise l'interaction. De même, le décodage de la réception d'une interaction par une entité est équivalent à l'opération inverse du codage d'une entité par un ou plusieurs îlots fonctionnels.

Le codage et le décodage peuvent ainsi être définis dans l'activité d'architecture fonctionnelle statique étiquetée par un cercle plein dans la description de l'activité d'architecture fonctionnelle dynamique de la **Figure 130**.

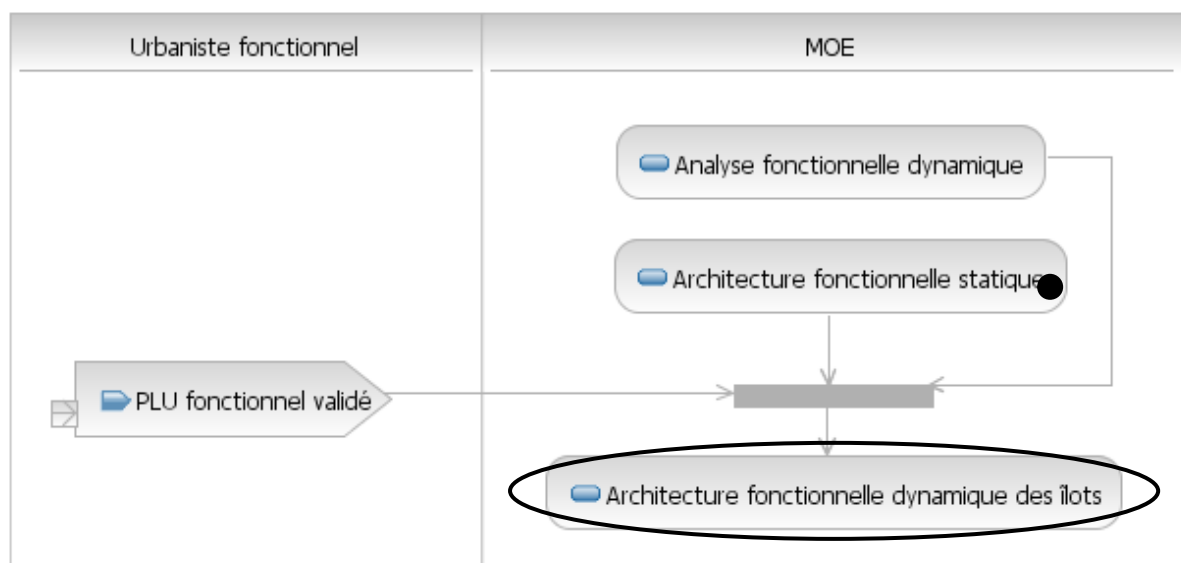


Figure 130 : Codage, décodage et activité d'architecture fonctionnelle dynamique des îlots

Ce lien avec l'activité d'architecture fonctionnelle statique permet un rapprochement avec l'ingénierie des modèles mise en œuvre dans la démarche EA4UP.

5.3.2 Codage d'information et approche MDA

La **Figure 131** représente la description du codage d'un système de communication par une transformation de modèle associée à l'activité d'architecture fonctionnelle statique de la démarche EA4UP (cf. §4.2.3.1).

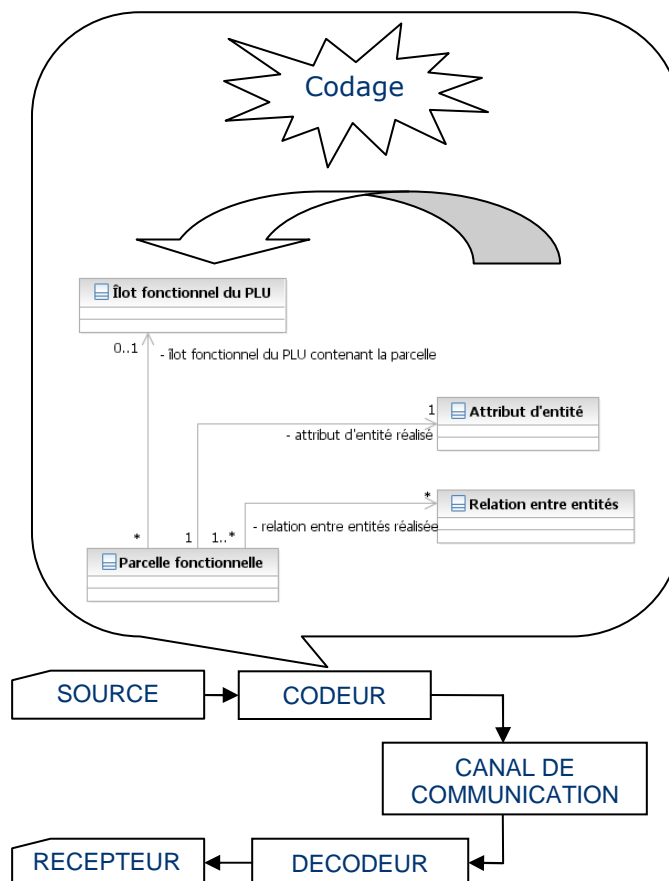


Figure 131 : Codage et approche MDA

La transmission par le canal de communication, soit le PLU fonctionnel, est ainsi rendu possible par le codage des entitésinstanciées par les interactions. Les interactions entre îlots du PLU fonctionnel sont en effet déduites du codage des attributs des entités ou du codage des relations entre les entités de l'interaction (cf. Règle d'urbanisme – Alignement fonctionnel du §5.1.1).

L'illustration choisie pour l'alignement fonctionnel est celle des interactions entre entités du scénario *Sc-Filtrage négatif* présenté dans le §4.2.1.2. Les interactions entre entités sont celles du diagramme de séquence de la **Figure 72** et le résultat du codage est représenté dans le Tableau 13. Afin que l'illustration soit plus pertinente, le PLU fonctionnel est le modèle d'îlots fonctionnels extrait de la **Figure 90**. La **Figure 132** reprend les informations décrites dans ces diagrammes et ce tableau.

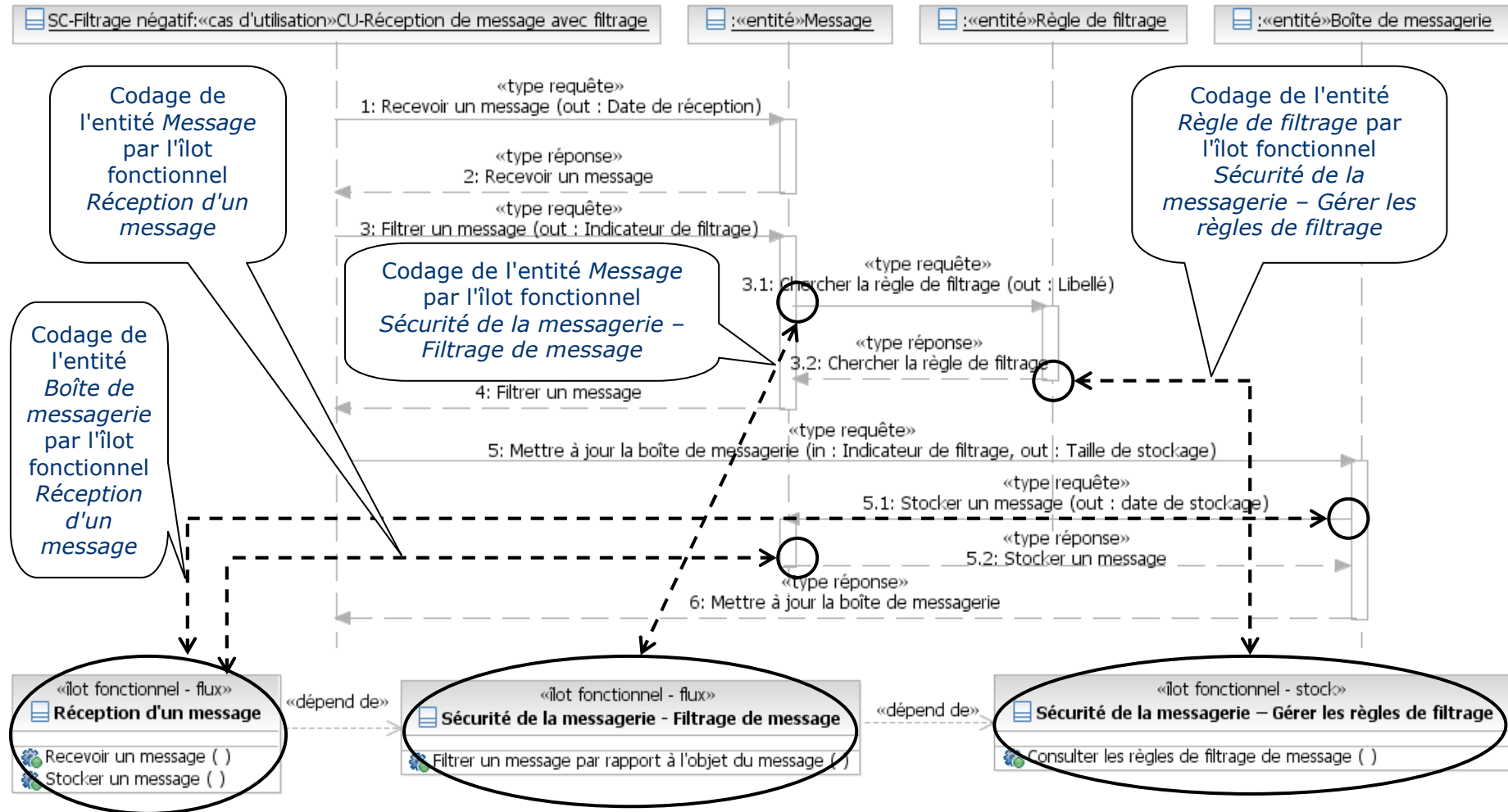


Figure 132 : Illustration du codage de l'alignement fonctionnel

5.3.3 Décodage d'information et approche MDA

La **Figure 133** représente la description du décodage d'un système de communication par une transformation de modèle déduite du codage de l'activité d'architecture fonctionnelle statique de la démarche EA4UP (cf. §4.2.3.1).

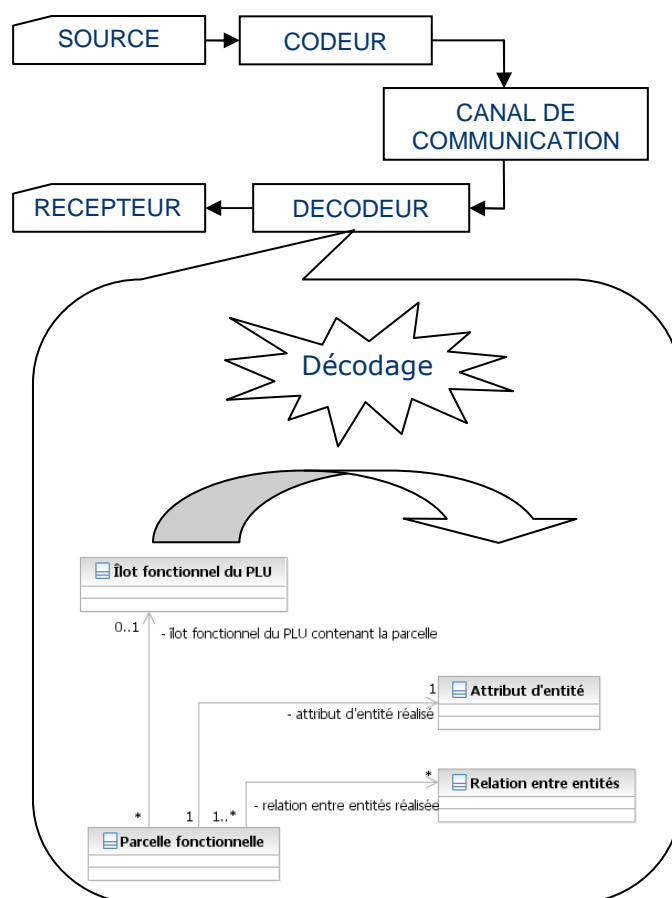


Figure 133 : Décodage et approche MDA

L'opération de décodage est l'opération inverse du codage permettant d'associer des attributs de l'entité cible de l'interaction aux îlots fonctionnels du PLU. L'association est conforme à la Règle d'urbanisme – Alignement fonctionnel du §5.1.1. Le décodage est une opération de vérification de la perte d'information pour atteindre l'entité cible. Dans l'illustration de la **Figure 124**, le décodage consiste à vérifier que les îlots fonctionnels codant l'entité *E2* sont accessibles par une voie du PLU fonctionnel à partir des îlots fonctionnels codant l'entité *E1*.

Le canal de communication établit l'amplitude de la perte d'informations lorsque l'entité cible doit être atteinte après le décodage.

Le décodage pour l'alignement fonctionnel, illustré avec le scénario SC-Filtrage négatif, est représenté dans la **Figure 134**.

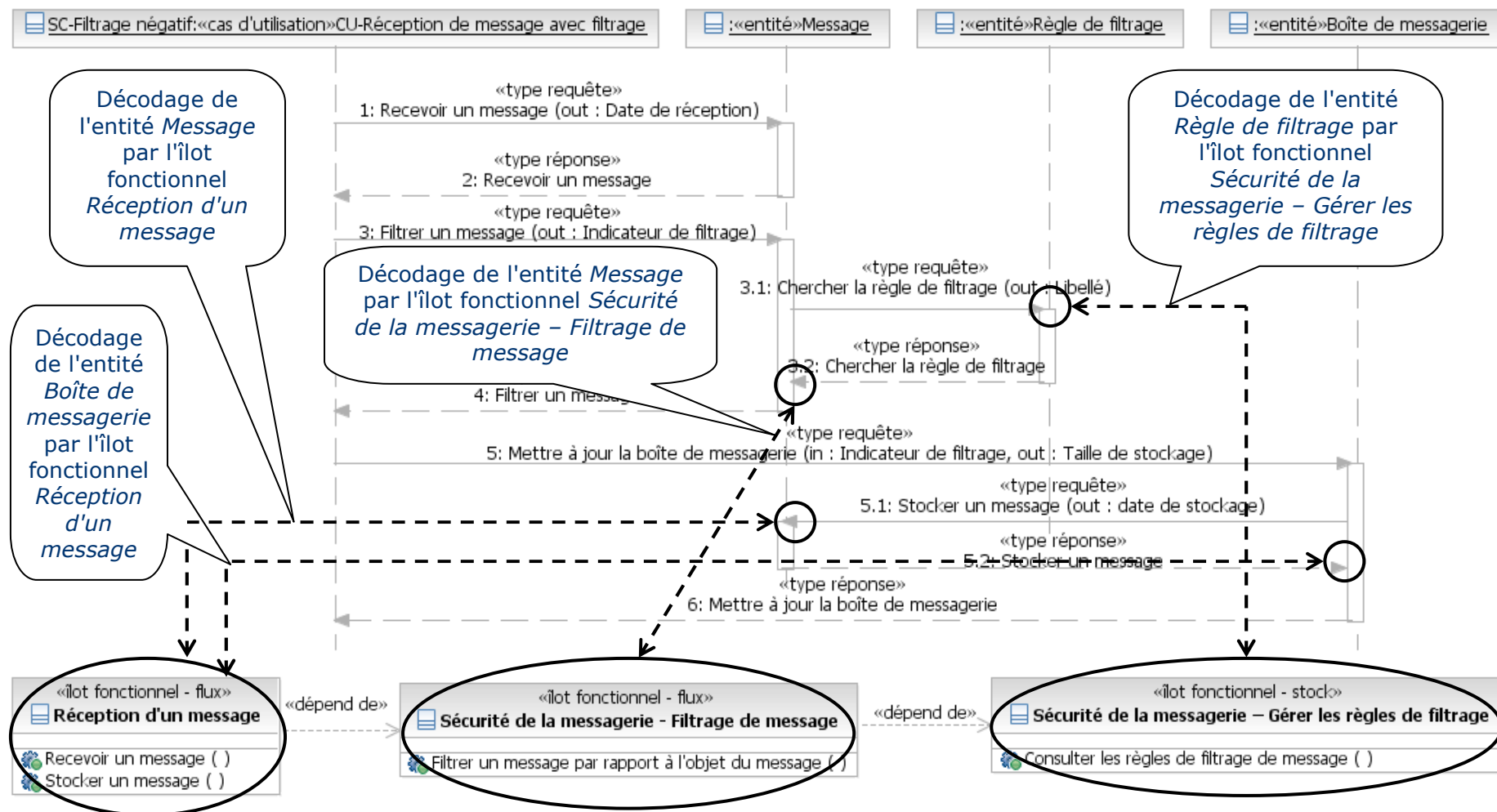


Figure 134 : Illustration du décodage de l'alignement fonctionnel

5.4 PLU fonctionnel et canal de communication

5.4.1 Canal de communication et approche MDE

Le canal de communication doit permettre la transmission de l'information codée vers l'information à décoder. Cette transmission est conditionnée par le PLU fonctionnel conçu par les urbanistes fonctionnels.

Dans le diagramme de séquence extrait de la **Figure 124**, ceci signifie que l'existence de voies du PLU fonctionnel est nécessaire :

- entre les îlots fonctionnels *I1* et *I2*
- de l'îlot fonctionnel *I2* vers l'îlot fonctionnel *I3*
- de l'îlot fonctionnel *I1* vers l'îlot fonctionnel *I3*

pour transmettre l'information codée. Ce conditionnement par le PLU fonctionnel est représenté dans la **Figure 135**.

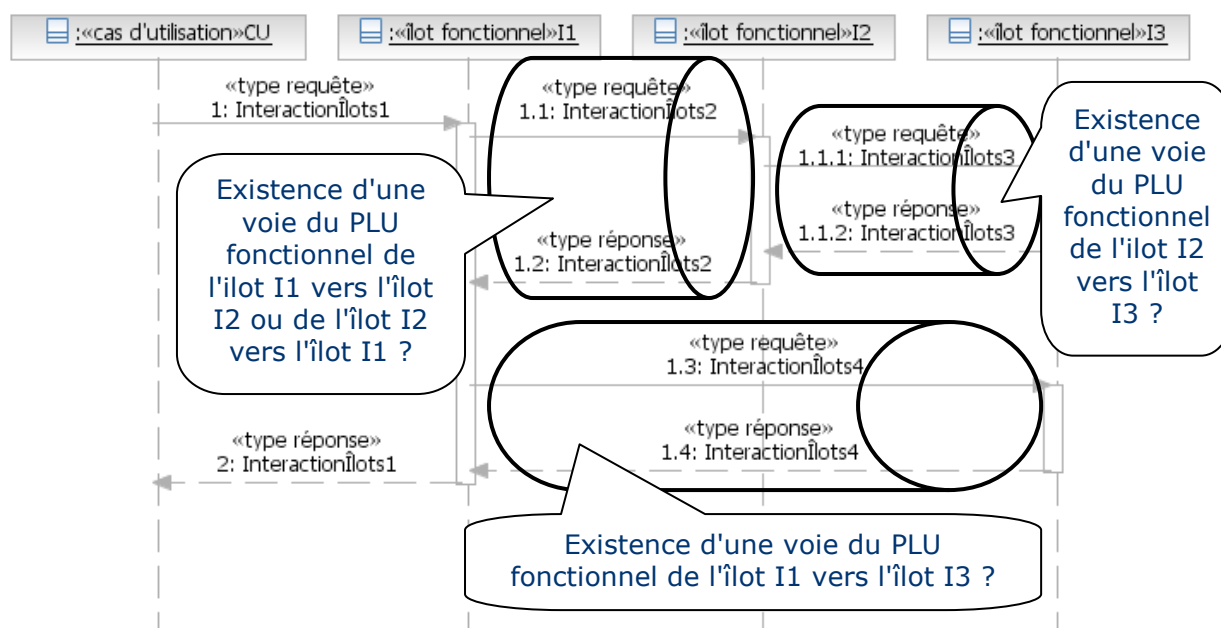


Figure 135 : Canal de communication et PLU fonctionnel

Les îlots *I1* et *I2* codent en effet la même entité *E1*. De plus, l'interaction *InteractionEntités2* de type requête de l'entité *E1* vers l'entité *E2* est alignée avec :

- l'interaction de type requête *InteractionÎlots3* de l'îlot fonctionnel *I2* vers l'îlot fonctionnel *I3* et
- l'interaction de type requête *InteractionÎlots4* de l'îlot fonctionnel *I1* vers l'îlot fonctionnel *I3*.

5.4.2 Niveaux du PLU et loi de transition

Le canal de communication est discret dans le contexte de la transmission liée aux interactions entre îlots fonctionnels alignées avec des interactions entre entités. Il peut donc être caractérisé par un alphabet d'entrée X_α et un alphabet de sortie Y_α .

X_α est l'alphabet des événements réalisés par la variable aléatoire X suite au codage des interactions entre entités. L'alphabet X_α contient donc les îlots fonctionnels codant les entités à l'origine des interactions spécifiées lors de l'analyse fonctionnelle dynamique. Y_α est l'alphabet des événements réalisés par la variable aléatoire Y avant le décodage des interactions entre îlots. L'alphabet Y_α contient les îlots fonctionnels à décoder en entités cibles de ces interactions.

Dans l'illustration du filtrage d'un message, les alphabets sont immédiatement déduits des illustrations du codage et du décodage :

- d'après la **Figure 132**, $X_\alpha = \{\text{Réception d'un message, Sécurité de la messagerie - Filtrage de message, Sécurité de la messagerie - Gérer les règles de filtrage}\}$,
- d'après la **Figure 134**, $Y_\alpha = \{\text{Réception d'un message, Sécurité de la messagerie - Filtrage de message, Sécurité de la messagerie - Gérer les règles de filtrage}\}$.

Définition : Le canal de communication est caractérisé par un alphabet d'entrée X_α , un alphabet de sortie Y_α et une loi de transition $P_{Y|X}$ qui caractérise le PLU fonctionnel. La matrice stochastique $M_{Y|X}$ qui représente la loi de transition $P_{Y|X}$ est constituée d'éléments $m_{i,j}$ (i^o ligne et j^o colonne) tels que :

$$m_{i,j} = p(I_j | I_i) \text{ avec } I_i \in X_\alpha, I_j \in Y_\alpha$$

Définition : Les fonctions d'appartenance d'un îlot fonctionnel à un élément d'un des trois niveaux du PLU (îlot, quartier, zone) sont définies de la façon suivante :

- $\forall I$ un îlot fonctionnel, alors $f_i(I) = I$, soit la fonction identité pour les îlots fonctionnels,
- $\forall I$ un îlot fonctionnel, $\exists Q$ quartier fonctionnel tel que $f_q(I) = Q$ si $I \in Q$,
- $\forall I$ un îlot fonctionnel, $\exists Z$ zone fonctionnelle tel que $f_z(I) = Z$ si $I \in Z$.



Figure 136 : Illustration du PLU fonctionnel

Pour illustrer les fonctions d'appartenance, supposons qu'il existe deux quartiers fonctionnels dans le PLU fonctionnel appelé dans la **Figure 136** :

- *Gestion de la sécurité* de la zone fonctionnelle *Transverse*,
- *Gestion de message* de la zone fonctionnelle *Messagerie*.

Le PLU fonctionnel est alors défini par les fonctions d'appartenance suivantes :

- $f_q(\text{Réception d'un message}) = \text{Gestion de message}$,
- $f_q(\text{Sécurité de la messagerie - Filtrage de message}) = f_q(\text{Sécurité de la messagerie - Gérer les règles de filtrage}) = \text{Gestion de la sécurité}$,
- $f_z(\text{Réception d'un message}) = \text{Messagerie}$,
- $f_z(\text{Sécurité de la messagerie - Filtrage de message}) = f_q(\text{Sécurité de la messagerie - Gérer les règles de filtrage}) = \text{Transverse}$.

Illustration 5-1

Les voies du PLU fonctionnel sont déduites des voies fonctionnelles entre îlots, il existe donc dans l'illustration 5-1 :

- une voie du quartier fonctionnel *Gestion de message* vers le quartier fonctionnel *Gestion de la sécurité*,
- une voie de la zone fonctionnelle *Messagerie* vers la zone fonctionnelle *Transverse*.

De plus, pour chaque élément du niveau îlot, du niveau quartier ou du niveau zone du PLU fonctionnel, une voie est ajoutée de l'élément vers lui-même.

La probabilité de l'interaction de l'îlot fonctionnel I_i vers l'îlot fonctionnel I_j est estimée à l'aide de la variable aléatoire L qui réalise les événements représentant l'appartenance d'une voie à un des trois niveaux du PLU fonctionnel ($L = \hat{i}$ au niveau îlot, $L = q$ au niveau quartier, $L = z$ au niveau zone).

L'estimation a priori de $P(L)$ est telle que :

$$\begin{aligned} - \quad p(L = \hat{i}) &= \frac{\text{Card}(E_{v\hat{i}}(PLU))}{D} \\ - \quad p(L = q) &= \frac{\text{Card}(E_{vq}(PLU))}{D} \\ - \quad p(L = z) &= \frac{\text{Card}(E_{vz}(PLU))}{D} \end{aligned}$$

avec

- $E_{v\hat{i}}(PLU) = \{\text{voies du PLU entre îlots fonctionnels}\}$
- $E_{vq}(PLU) = \{\text{voies du PLU entre quartiers fonctionnels}\}$
- $E_{vz}(PLU) = \{\text{voies du PLU entre zones fonctionnelles}\}$
- $D = \text{Card}(E_{v\hat{i}}(PLU)) + \text{Card}(E_{vq}(PLU)) + \text{Card}(E_{vz}(PLU))$

D'après l'illustration 5-1 sur les fonctions d'appartenance, les cardinalités des ensembles de voies du PLU fonctionnel sont les suivantes :

- $\text{Card}(E_{v\hat{i}}(PLU)) = 5,$
- $\text{Card}(E_{vq}(PLU)) = 3,$
- $\text{Card}(E_{vz}(PLU)) = 3,$

$$\Rightarrow D = 11 \Rightarrow$$

$$\begin{aligned} - \quad p(L = \hat{i}) &= \frac{5}{11} \\ - \quad p(L = q) &= \frac{3}{11} \\ - \quad p(L = z) &= \frac{3}{11} \end{aligned}$$

Proposition : La loi de transition $P_{Y|X}$ est estimée à partir de la variable aléatoire L de la façon suivante :

$$P(Y|X) = \sum_L P(L) * P(Y|f_L(Y), L) * P(f_L(Y)|f_L(X), L)$$

Équation 5-2

La matrice stochastique $M_{Y|X}$ représentant la loi de transition $P_{Y|X}$ peut alors être évaluée pour toutes les lettres I_i de l'alphabet d'entrée X_α et pour toutes les lettres I_j de l'alphabet de sortie Y_α :

$$p(I_j|I_i) = \alpha_i \left(\frac{Ind_{v\hat{i}}(I_i, I_j)}{Card(E_{v\hat{i}}(I_i))} \right) + \alpha_q \left(\frac{1}{Card(E_i(f_q(I_j)))} * \frac{Ind_{vq}(f_q(I_i), f_q(I_j))}{Card(E_{vq}(f_q(I_i)))} \right) + \alpha_z \left(\frac{1}{Card(E_i(f_z(I_j)))} * \frac{Ind_{vz}(f_z(I_i), f_z(I_j))}{Card(E_{vz}(f_z(I_i)))} \right)$$

Équation 5-3

avec

- $\alpha_i = p(L = \hat{i})$,
- $\alpha_q = p(L = q)$,
- $\alpha_z = p(L = z)$,
- $Ind_{v\hat{i}}(I_i, I_j)$ = indicateur booléen de l'existence d'une voie du PLU fonctionnel entre îlots de I_i vers I_j ,
- $Ind_{vq}(Q_i, Q_j)$ = indicateur booléen de l'existence d'une voie du PLU fonctionnel entre quartiers de Q_i vers Q_j ,
- $Ind_{vz}(Z_i, Z_j)$ = indicateur booléen de l'existence d'une voie du PLU fonctionnel entre zones de Z_i vers Z_j ,
- $E_{v\hat{i}}(I_i) = \{\text{voies du PLU fonctionnel entre îlots ayant pour origine } I_i\}$,
- $E_{vq}(Q_i) = \{\text{voies du PLU fonctionnel entre quartiers ayant pour origine } Q_i\}$,
- $E_{vz}(Z_i) = \{\text{voies du PLU fonctionnel entre zones ayant pour origine } Z_i\}$,
- $E_i(Q_i) = \{I_k; Q_i = f_q(I_k)\}$ pour un quartier fonctionnel Q_i ou
 $E_i(Z_i) = \{I_k; Z_i = f_z(I_k)\}$ pour une zone fonctionnelle Z_i .

Démonstration :

Soit $p_i(I_j|I_i) = p(Y = I_j | f_i(Y = I_j), L = \hat{i}) * p(f_c(Y = I_j) | f_c(X = I_i), L = c) = p(I_j|I_i)$

puisque $p(Y = I_j | f_i(Y = I_j), L = \hat{i}) = 1$ et $p(f_c(Y = I_j) | f_c(X = I_i), L = c) = p(I_j|I_i)$.

Or, par hypothèse, les dépendances entre îlots fonctionnels du PLU sont équiprobables :

$$\Rightarrow p_i(I_j|I_i) = \frac{Ind_{v\hat{i}}(I_i, I_j)}{Card(E_{v\hat{i}}(I_i))}$$

Soit $p_q(I_j|I_i) = p(Y = I_j | f_q(Y = I_j), L = q) * p(f_q(Y = I_j) | f_q(X = I_i), L = q)$

$$\Rightarrow p_q(I_j|I_i) = p(I_j|f_q(I_i)) * p(f_q(I_j) | f_q(I_i))$$

Par hypothèse, les dépendances entre quartiers fonctionnels du PLU d'une part, et l'appartenance d'îlots à un même quartier d'autre part sont équiprobables :

$$\Rightarrow P_q(I_j|I_i) = \frac{1}{\text{Card}(E_c(f_q(I_j)))} * \frac{I_{dq}(f_q(I_i), f_q(I_j))}{\text{Card}(E_{dq}(f_q(I_i)))}$$

Soit $p_z(I_j|I_i) = p(Y = I_j | f_z(Y = I_j), L = z) * p(f_z(Y = I_j) | f_z(X = I_i), L = z)$

$$\Rightarrow p_z(I_j|I_i) = p(I_j | f_z(I_j)) * p(f_z(I_j) | f_z(I_i))$$

De la même façon, par hypothèse, les dépendances entre zones fonctionnelles du PLU sont équiprobables ainsi que l'appartenance d'îlots à une même zone :

$$\Rightarrow p_z(I_j|I_i) = \frac{1}{\text{Card}(E_c(f_z(I_j)))} * \frac{I_{dz}(f_z(I_i), f_z(I_j))}{\text{Card}(E_{dz}(f_z(I_i)))}$$

L'application de l'Équation 5-2 aux lettres des alphabets X_α et Y_α implique alors l'Équation 5-3.

Dans l'illustration 5-1, la matrice stochastique est évaluée à partir des alphabets $X_\alpha = Y_\alpha = \{I_1 = \text{Réception d'un message}, I_2 = \text{Sécurité de la messagerie} - \text{Filtrage de message}, I_3 = \text{Sécurité de la messagerie} - \text{Gérer les règles de filtrage}\}$.

D'après l'illustration de l'estimation de $P(L)$:

- $\alpha_i = \frac{5}{11}$,
- $\alpha_q = \frac{3}{11}$,
- $\alpha_z = \frac{3}{11}$.

D'après l'Équation 5-3, les éléments de la matrice stochastique sont les suivants :

- $p(I_1|I_1) = \left(\alpha_i * \frac{1}{2}\right) + \left(\alpha_q * \frac{1}{1} * \frac{1}{2}\right) + \left(\alpha_z * \frac{1}{1} * \frac{1}{2}\right)$
- $p(I_2|I_1) = \left(\alpha_i * \frac{1}{2}\right) + \left(\alpha_q * \frac{1}{2} * \frac{1}{2}\right) + \left(\alpha_z * \frac{1}{2} * \frac{1}{2}\right)$
- $p(I_3|I_1) = \left(\alpha_i * \frac{0}{2}\right) + \left(\alpha_q * \frac{1}{2} * \frac{1}{2}\right) + \left(\alpha_z * \frac{1}{2} * \frac{1}{2}\right)$
- $p(I_1|I_2) = \left(\alpha_i * \frac{0}{2}\right) + \left(\alpha_q * \frac{1}{1} * \frac{0}{1}\right) + \left(\alpha_z * \frac{1}{1} * \frac{0}{1}\right)$
- $p(I_2|I_2) = \left(\alpha_i * \frac{1}{2}\right) + \left(\alpha_q * \frac{1}{2} * \frac{1}{1}\right) + \left(\alpha_z * \frac{1}{2} * \frac{1}{1}\right)$
- $p(I_3|I_2) = \left(\alpha_i * \frac{1}{2}\right) + \left(\alpha_q * \frac{1}{2} * \frac{1}{1}\right) + \left(\alpha_z * \frac{1}{2} * \frac{1}{1}\right)$
- $p(I_1|I_3) = \left(\alpha_i * \frac{0}{1}\right) + \left(\alpha_q * \frac{1}{1} * \frac{0}{1}\right) + \left(\alpha_z * \frac{1}{1} * \frac{0}{1}\right)$
- $p(I_2|I_3) = \left(\alpha_i * \frac{0}{1}\right) + \left(\alpha_q * \frac{1}{2} * \frac{1}{1}\right) + \left(\alpha_z * \frac{1}{2} * \frac{1}{1}\right)$
- $p(I_3|I_3) = \left(\alpha_i * \frac{1}{1}\right) + \left(\alpha_q * \frac{1}{2} * \frac{1}{1}\right) + \left(\alpha_z * \frac{1}{2} * \frac{1}{1}\right)$

$$\Rightarrow M_{Y|X} = \begin{pmatrix} \frac{1}{2} & \frac{8}{22} & \frac{3}{22} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{3}{11} & \frac{8}{11} \end{pmatrix}$$

Le canal de communication représenté par le PLU fonctionnel est donc défini par l'alphabet d'entrée X_α , l'alphabet de sortie Y_α et par cette matrice stochastique $M_{Y|X}$.

5.5 Mesure d'alignement fonctionnel et information mutuelle moyenne

La loi de transition $P_{Y|X}$ estimée par l'Équation 5-3 permet d'évaluer l'entropie conditionnelle de la variable aléatoire Y sachant X . Cette entropie conditionnelle est le paramètre principal de l'information mutuelle moyenne (cf. §5.1.3) qui caractérise l'alignement de la vue fonctionnelle d'un système avec l'EA fonctionnelle du SI auquel il appartient.

5.5.1 Choix de l'information mutuelle moyenne

L'information mutuelle moyenne est la mesure d'alignement choisie entre la vue fonctionnelle d'un système et une EA fonctionnelle. Cette mesure permet d'évaluer la quantité d'information apportée par la cible d'une interaction entre entités sur l'origine de l'interaction. La quantité d'information est transmise par le canal de communication. Dans le cas d'interactions entre entités, la perte d'information n'est pas totale lorsque les interactions entre îlots fonctionnels alignés avec ces interactions respectent la Règle d'urbanisme – Interactions et voies (cf. §3.2) avec des voies du PLU fonctionnel. Lorsqu'il n'y a pas de voies du PLU fonctionnel respectant cette règle, la perte d'information est complète.

Un intérêt majeur de l'information mutuelle moyenne est de représenter l'amélioration de la cohérence entre l'analyse dynamique des exigences fonctionnelles et le LUP fonctionnel. L'amélioration de la cohérence signifie une relation plus forte entre l'îlot fonctionnel à l'origine d'une interaction, qui est alignée avec une interaction entre entités, et l'îlot fonctionnel cible de cette interaction. Or, la force de cette relation est caractérisée par la dépendance au sens UML entre ces îlots fonctionnels à chaque niveau du LUP. La relation est plus forte pour une voie entre îlots que pour une voie entre quartiers, elle-même plus forte qu'une voie entre zones.

Une meilleure cohérence de l'analyse dynamique des exigences fonctionnelles avec le LUP fonctionnel est ainsi équivalente à minimiser, pour un ensemble *Int* d'interactions entre entités, l'entropie conditionnelle de Y connaissant X . Ceci revient d'après le diagramme de Venn de la **Figure 126** à maximiser l'information mutuelle moyenne $I(X(Int);Y(Int))$.

Pour être pertinente, l'étude de la variation de l'information mutuelle moyenne nécessite un nombre significatif d'interactions entre entités. Ce nombre significatif étant induit par l'approche dynamique choisie pour la démarche de développement de système EA4UP (cf. §4.1.2), le choix de l'information mutuelle moyenne est justifié.

D'après la définition de la mesure d'alignement MAFS décrite dans l'Équation 5-1 pour des interactions de l'ensemble Int , l'ensemble des interactions entre entités spécifiées en analyse fonctionnelle dynamique, la mesure MAFS est définie de la manière suivante :

$$MAFS(Int) = I(X(Int); Y(Int)) = H(Y(Int)) - H(Y(Int)|X(Int))$$

avec, si x est un évènement réalisé par X et y est un évènement réalisé par Y ,

$$H(Y) = - \sum_y p(y) \log_2(p(y))$$

Équation 5-4

$$H(Y|X) = - \sum_{x,y} p(x,y) \log_2(p(y|x))$$

Équation 5-5

5.5.2 Formule de l'information mutuelle moyenne pour l'alignement fonctionnel

Proposition : La mesure d'alignement fonctionnel MAFS de la vue fonctionnelle d'un système avec l'EA fonctionnelle du SI auquel appartient le système est pour les interactions int , appartenant à l'ensemble Int des interactions entre îlots fonctionnels alignées avec des interactions entre entités, telle que :

$$MAFS(Int) = - \left(\prod_{j=1}^{jn} \frac{Card(E_{ir}(I_{ji}))}{\sum_{I_i \in Y_\alpha} Card(E_{ir}(I_i))} * \log_2 \left(\prod_{j=1}^{jn} \frac{Card(E_{ir}(I_{ji}))}{\sum_{I_i \in Y_\alpha} Card(E_{ir}(I_i))} \right) \right) +$$

$$\left(\prod_{int=1}^{int_k} \left(\alpha_i \left(\frac{Ind_{v\hat{i}}(I_i, I_j)}{Card(E_{v\hat{i}}(I_i))} \right) + \alpha_q \left(\frac{1}{Card(E_{\hat{i}}(f_q(I_j)))} * \frac{Ind_{vq}(f_q(I_i), f_q(I_j))}{Card(E_{vq}(f_q(I_i)))} \right) + \right.$$

$$\left. \alpha_z \left(\frac{1}{Card(E_{\hat{i}}(f_z(I_j)))} * \frac{Ind_{vz}(f_z(I_i), f_z(I_j))}{Card(E_{vz}(f_z(I_i)))} \right) \right) * \left(\prod_{i=1}^{in} \frac{Card(E_{i\hat{e}}(I_{ii}))}{\sum_{I_i \in X_\alpha} Card(E_{i\hat{e}}(I_i))} \right) * \log_2 \left(\alpha_i \left(\frac{Ind_{v\hat{i}}(I_i, I_j)}{Card(E_{v\hat{i}}(I_i))} \right) + \alpha_q \left(\frac{1}{Card(E_{\hat{i}}(f_q(I_j)))} * \frac{Ind_{vq}(f_q(I_i), f_q(I_j))}{Card(E_{vq}(f_q(I_i)))} \right) + \right.$$

$$\left. \alpha_z \left(\frac{1}{Card(E_{\hat{i}}(f_z(I_j)))} * \frac{Ind_{vz}(f_z(I_i), f_z(I_j))}{Card(E_{vz}(f_z(I_i)))} \right) \right)$$

Équation 5-6

avec :

- $E_{ir}(I_j) = \{int \in Int \text{ tel que } I_j \in Y_\alpha \text{ soit la cible de l'interaction } int \text{ entre îlots fonctionnels alignée avec une interaction entre entités de } Sc\}$,
- $E_{i\hat{e}}(int) = \{(I_i, I_j), I_i \in X_\alpha, I_j \in Y_\alpha \text{ et tel que } I_i \text{ soit à l'origine de l'interaction } int \in Int \text{ et } I_j \text{ à la cible de cette interaction } int\}$,
- $\alpha_i = p(L = \hat{i})$,
- $\alpha_q = p(L = q)$,

- $\alpha_z = p(L = z)$,
- $Ind_{vi}(I_i, I_j)$ = indicateur booléen de l'existence d'une voie du PLU fonctionnel entre îlots de I_i vers I_j ,
- $Ind_{vq}(Q_i, Q_j)$ = indicateur booléen de l'existence d'une voie du PLU fonctionnel entre quartiers de Q_i vers Q_j ,
- $Ind_{vz}(Z_i, Z_j)$ = indicateur booléen de l'existence d'une voie du PLU fonctionnel entre zones de Z_i vers Z_j ,
- $E_{vi}(I_i) = \{\text{voies du PLU fonctionnel entre îlots ayant pour origine } I_i\}$,
- $E_{vq}(Q_i) = \{\text{voies du PLU fonctionnel entre quartiers ayant pour origine } Q_i\}$,
- $E_{vz}(Z_i) = \{\text{voies du PLU fonctionnel entre zones ayant pour origine } Z_i\}$,
- $E_i(Q_i) = \{I_k; Q_i = f_q(I_k)\}$ pour un quartier fonctionnel Q_i ou $E_i(Z_i) = \{I_k; Z_i = f_z(I_k)\}$ pour une zone fonctionnelle Z_i .

Démonstration :

La démonstration est en trois parties : le calcul de $H(Y)$, le calcul de $H(Y|X)$ et le calcul de MAFS.

Calcul de $H(Y)$

Soit l'ensemble Sc des scénarios décrits lors de la collecte des exigences fonctionnelles dynamiques du système. L'estimation de la probabilité des événements y issus du décodage des îlots fonctionnels, cibles d'interactions entre îlots alignées avec les interactions entre entités de chaque scénario s de Sc , est telle que :

$$p(y) = \sum_{s \in Sc} p(y|s) * p(s)$$

En première approche, les scénarios sont équiprobables. Un seul événement peut donc être pris en compte à l'issue du décodage puisqu'il ne dépend pas du scénario. L'ensemble des interactions des scénarios de Sc est donc pris en compte sans tenir compte du scénario où chaque interaction est instanciée.

Soit Int l'ensemble des interactions tel que $Int = \{int_i; int_i \text{ soit une interaction entre îlots fonctionnels alignée avec une interaction entre entités de } Sc\}$.

Si l'évènement $y = (I_{j1})_{int_1} \dots (I_{jn})_{int_k}$ est réalisé par le processus aléatoire Y alors,

$$p(y) = p((I_{j1})_{int_1} \dots (I_{jn})_{int_k}) = p((I_{j1})_{int_1}) * \dots * p((I_{jn})_{int_k})$$

puisque les réceptions d'interactions sont indépendantes.

Si $E_{ir}(I_j) = \{int, \text{ tel que } I_j \in Y_\alpha \text{ soit la cible de l'interaction } int \text{ entre îlots fonctionnels alignée avec une interaction entre entités de } Sc\}$.

$$\Rightarrow \forall int \in Int, p((I_{ji})_{int}) = \frac{Card(E_{ir}(I_{ji}))}{\sum_{I_i \in Y_\alpha} Card(E_{ir}(I_i))} \Rightarrow p(y) = \prod_{ji=j1}^{jn} \frac{Card(E_{ir}(I_{ji}))}{\sum_{I_i \in Y_\alpha} Card(E_{ir}(I_i))}$$

Dans cette approche où un seul événement y , lié à l'ensemble des interactions, est réalisé par Y , d'après Équation 5-4 :

$$H(Y) = - \left(\prod_{ji=j1}^{jn} \frac{Card(E_{ir}(I_{ji}))}{\sum_{I_i \in Y_\alpha} Card(E_{ir}(I_i))} * \log_2 \left(\prod_{ji=j1}^{jn} \frac{Card(E_{ir}(I_{ji}))}{\sum_{I_i \in Y_\alpha} Card(E_{ir}(I_i))} \right) \right)$$

Équation 5-7

Calcul de $H(Y|X)$

Si l'évènement $x = (I_{i1})_{int_1} \dots (I_{in})_{int_k}$ est réalisé par le processus aléatoire X et si l'évènement $y = (I_{j1})_{int_1} \dots (I_{jn})_{int_k}$ est réalisé par le processus aléatoire Y alors,

$$p(y|x) = p\left((I_{j1})_{int_1} \dots (I_{jn})_{int_k} \mid (I_{i1})_{int_1} \dots (I_{in})_{int_k}\right) \Rightarrow p(y|x) = p\left((I_{j1})_{int_1} \mid (I_{i1})_{int_1}\right) * \dots * p\left((I_{jn})_{int_k} \mid (I_{in})_{int_k}\right)$$

puisque les évènements représentés par des couples d'îlots fonctionnels d'une même interaction sont indépendants les uns des autres.

Soit $E_{ii}(int) = \{(I_i, I_j), I_i \in X_\alpha, I_j \in Y_\alpha \text{ et tel que } I_i \text{ soit à l'origine de l'interaction } int \text{ et } I_j \text{ à la cible de l'interaction } int\}$. D'après l'Équation 5-3 :

$$p(y|x) = \prod_{int=int_1}^{int_k} \left((I_i, I_j) \in E_{ii}(int) \right) \left(\alpha_i \left(\frac{Ind_{v_i}(I_i, I_j)}{Card(E_{v_i}(I_i))} \right) + \alpha_q \left(\frac{1}{Card(E_i(f_q(I_j)))} * \frac{Ind_{v_q}(f_q(I_i), f_q(I_j))}{Card(E_{v_q}(f_q(I_i)))} \right) + \alpha_z \left(\frac{1}{Card(E_i(f_z(I_j)))} * \frac{Ind_{v_z}(f_z(I_i), f_z(I_j))}{Card(E_{v_z}(f_z(I_i)))} \right) \right)$$

Équation 5-8

Comme pour l'évènement y réalisé par Y , un seul évènement x lié à l'ensemble des interactions est réalisé par X .

Si l'évènement $x = (I_{i1})_{int_1} \dots (I_{in})_{int_k}$ est réalisé par le processus aléatoire X alors,

$$p(x) = p\left((I_{i1})_{int_1} \dots (I_{in})_{int_k}\right) = p\left((I_{i1})_{int_1}\right) * \dots * p\left((I_{in})_{int_k}\right)$$

puisque les émissions d'interactions sont indépendantes.

$E_{ie}(I_j) = \{int, \text{ tel que } I_j \in Y_\alpha \text{ soit l'origine de l'interaction } int \text{ entre îlots fonctionnels alignée avec une interaction entre entités de } Sc\}$

$$\Rightarrow \forall int \in Int, p\left((I_{ii})_{int}\right) = \frac{Card(E_{ie}(I_{ii}))}{\sum_{I_i \in X_\alpha} Card(E_{ie}(I_i))}$$

$$\Rightarrow p(x) = \prod_{ii=i1}^{in} \frac{Card(E_{ie}(I_{ii}))}{\sum_{I_i \in X_\alpha} Card(E_{ie}(I_i))}$$

Équation 5-9

Comme $p(x, y) = p(y|x) * p(x)$, d'après l'Équation 5-5, l'Équation 5-8 et l'Équation 5-9 :

$$H(Y|X) = - \left(\prod_{\text{int}=\text{int}_1: (I_i, I_j) \in E_{ii}(\text{int})}^{\text{int}_k} \left(\alpha_i \left(\frac{\text{Ind}_{v_i}(I_i, I_j)}{\text{Card}(E_{v_i}(I_i))} \right) + \alpha_q \left(\frac{1}{\text{Card}(E_{\hat{i}}(f_q(I_j)))} * \frac{\text{Ind}_{v_q}(f_q(I_i), f_q(I_j))}{\text{Card}(E_{v_q}(f_q(I_i)))} \right) + \right. \right. \\ \left. \left. \alpha_z \left(\frac{1}{\text{Card}(E_{\hat{i}}(f_z(I_j)))} * \frac{\text{Ind}_{v_z}(f_z(I_i), f_z(I_j))}{\text{Card}(E_{v_z}(f_z(I_i)))} \right) \right) * \left(\prod_{ii=i1}^{\text{in}} \frac{\text{Card}(E_{i\hat{e}}(I_{ii}))}{\sum_{I_i \in X_\alpha} \text{Card}(E_{i\hat{e}}(I_i))} \right) \right) \\ \log_2 \left(\alpha_i \left(\frac{\text{Ind}_{v_i}(I_i, I_j)}{\text{Card}(E_{v_i}(I_i))} \right) + \alpha_q \left(\frac{1}{\text{Card}(E_{\hat{i}}(f_q(I_j)))} * \frac{\text{Ind}_{v_q}(f_q(I_i), f_q(I_j))}{\text{Card}(E_{v_q}(f_q(I_i)))} \right) + \right. \\ \left. \alpha_z \left(\frac{1}{\text{Card}(E_{\hat{i}}(f_z(I_j)))} * \frac{\text{Ind}_{v_z}(f_z(I_i), f_z(I_j))}{\text{Card}(E_{v_z}(f_z(I_i)))} \right) \right)$$

Équation 5-10

Calcul de MAFS

Puisque $MAFS(Int) = H(Y(Int)) - H(Y(Int)|X(Int))$, l'Équation 5-6 est immédiatement déduite de l'Équation 5-7 et de l'Équation 5-10.

Dans l'illustration du filtrage d'un message, l'ensemble Int est constituée d'après la **Figure 132** ou la **Figure 134** des interactions entre îlots fonctionnels suivantes :

- $Int = \{\text{Chercher la règle de filtrage - type requête, Chercher la règle de filtrage - type réponse, Stocker un message - type requête, Stocker un message - type réponse}\}.$

Pour les alphabets X_α et Y_α tels que $X_\alpha = Y_\alpha = \{I_1 = \text{Réception d'un message, } I_2 = \text{Sécurité de la messagerie - Filtrage de message, } I_3 = \text{Sécurité de la messagerie - Gérer les règles de filtrage}\}$ les événements x réalisé par X et y réalisé par Y sont d'après la **Figure 132** pour x et d'après la **Figure 134** pour y tels que :

- $x = (I_2)_{\text{Chercher la règle de filtrage type requête}} (I_3)_i \text{Chercher la règle de filtrage type réponse} (I_1)_{\text{Stocker un message type requête}}$
- $(I_1)_{\text{Stocker un message type réponse}}$
- $y = (I_3)_{\text{Chercher la règle de filtrage type requête}} (I_2)_i \text{Chercher la règle de filtrage type réponse} (I_1)_{\text{Stocker un message type requête}}$
- $(I_1)_{\text{Stocker un message type réponse}}$

Calcul de H(Y)

Les ensembles d'interactions reçues par chaque îlot de l'alphabet Y_α sont les suivants :

- $E_{ir}(I_1) = \{\text{Stocker un message} - \text{type requête}, \text{Stocker un message} - \text{type réponse}\},$
- $E_{ir}(I_2) = \{\text{Chercher la règle de filtrage} - \text{type réponse}\},$
- $E_{ir}(I_3) = \{\text{Chercher la règle de filtrage} - \text{type requête}\}.$

D'après l'Équation 5-7 :

$$H(Y) = -\left(\frac{1}{4} * \frac{1}{4} * \frac{2}{4} * \frac{2}{4} * \log_2\left(\frac{1}{4} * \frac{1}{4} * \frac{2}{4} * \frac{2}{4}\right)\right) \Rightarrow H(Y) = \frac{1}{8}$$

Calcul de H(Y|X)

Les ensembles d'interactions émises par chaque îlot de l'alphabet Y_α sont les suivants :

- $E_{ie}(I_1) = \{\text{Stocker un message} - \text{type requête}, \text{Stocker un message} - \text{type réponse}\},$
- $E_{ie}(I_2) = \{\text{Chercher la règle de filtrage} - \text{type requête}\},$
- $E_{ie}(I_3) = \{\text{Chercher la règle de filtrage} - \text{type réponse}\}.$

D'après l'Équation 5-9 :

$$p(x) = \frac{1}{4} * \frac{1}{4} * \frac{2}{4} * \frac{2}{4} = \frac{1}{64}$$

La matrice stochastique $M_{Y|X}$ évaluée dans le §5.4.2 est telle que

$$M_{Y|X} = \begin{pmatrix} \frac{1}{2} & \frac{8}{22} & \frac{3}{22} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{3}{11} & \frac{8}{11} \end{pmatrix}$$

Elle permet d'évaluer $p(y|x)$ pour chaque interaction d'après l'Équation 5-8 :

$$p(y|x) = \frac{1}{2} * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} = \frac{1}{16}$$

L'entropie conditionnelle est alors déduite de l'Équation 5-10 :

$$H(Y|X) = -\left(\frac{1}{64} * \frac{1}{16} * \log_2\left(\frac{1}{16}\right)\right) = \frac{1}{256}$$

Calcul de MAFS

La mesure MAFS de l'alignement fonctionnelle du service télécom avec l'EA fonctionnelle est donc :

$$MAFS(Int) = I(X(Int); Y(Int)) = H(Y(Int)) - H(Y(Int)|X(Int)) = \frac{1}{8} - \frac{1}{256} = \frac{31}{256}$$

L'ensemble *Int* des interactions a une information mutuelle moyenne très proche du maximum, soit l'entropie de la variable aléatoire de Y qui vaut $\frac{1}{8}$, puisque toutes les interactions sont conformes avec des voies du PLU fonctionnel entre îlots.

5.6 Conclusion

La mesure d'alignement fonctionnel entre la vue fonctionnelle d'un système et l'EA fonctionnelle du SI auquel appartient le système (cf. §5) permet de résoudre le problème de la granularité aléatoire des scénarios d'usage d'un système. L'information mutuelle moyenne offre une mesure pertinente du couplage lié aux interactions entre îlots fonctionnels réalisant le système.

La complexité de la définition de la mesure nécessite néanmoins un outillage. L'implémentation de la mesure d'alignement fonctionnel devrait être jointe à la démarche EA4UP afin d'assister l'architecte. Cette mesure est en effet un bon indicateur de l'écart entre le système et les préconisations des urbanistes fonctionnels. Un écart important peut être alors perçu comme une augmentation du coût de développement du fait d'une réutilisabilité moindre de composants applicatifs existants ou du fait d'une plus grande complexité de l'architecture du système. Cette mesure est ainsi un bon indicateur de la durabilité du SI dans lequel le système est développé.

6. Conclusion

L'objectif de la contribution de cette thèse est de proposer une assistance aux architectes, qu'ils soient des urbanistes fonctionnels du SI ou des architectes de systèmes de ce SI. Lors de la conception de l'architecture, l'assistance permet de donner de la lisibilité aux modèles d'architecture produits. Cette lisibilité facilite la réutilisation des composants applicatifs du SI par l'ensemble des architectes de ce SI. Or, une réutilisabilité assistée des composants applicatifs d'un SI est synonyme d'une évolution plus souple du SI et donc d'une meilleure durabilité de celui-ci.

Un premier facteur de durabilité du SI est la mise en avant de l'approche dynamique dans la démarche EA4UP. Dans une approche systémique, les unités du système conçus avec la démarche EA4UP sont les îlots fonctionnels ou les composants applicatifs. Chaque usage d'un système est représenté par des séquences d'interactions entre ces unités. De plus, pour les unités de type fonctionnel, chaque interaction peut être conçue de manière inductive grâce au modèle fonctionnel de référence du SI. Plus le modèle fonctionnel de référence est respecté par l'architecte du système, meilleure est la stabilité dynamique du SI dans sa globalité, et donc meilleure est sa durabilité. Si l'architecte préfère prendre des distances avec le modèle de référence, son choix de conception lui rend alors plus difficile la réutilisation d'éléments existants du SI. Le système ainsi conçu induirait alors des comportements inattendus néfastes à la durabilité du SI. Afin de renforcer cette stabilité dynamique, la prise en compte des interactions entre unités du système lors de son usage est prépondérante. Le modèle de données est directement déduit des séquences d'interactions entre unités. Une donnée est alors considérée comme un paramètre de sortie ou un paramètre d'entrée de chacune des interactions et non comme un élément intégré à un modèle des données conçu de manière indépendante de l'usage d'un système.

La conception des îlots fonctionnels est déduite de la stratégie de l'entreprise par les urbanistes du SI. A l'appui de cette conception d'expert, un algorithme bactériologique permet dans la démarche DA4EA de regrouper des îlots fonctionnels. Ce regroupement s'appuie sur le principe d'un couplage faible et d'une cohésion forte entre îlots fonctionnels regroupés. L'approche dynamique définie comme le premier facteur de durabilité d'un SI est prépondérante dans cet algorithme puisque le regroupement d'îlots fonctionnels dépend des différentes interactions entre les îlots d'une même zone fonctionnelle. Ces interactions sont spécifiées dans des scénarios cibles d'une zone fonctionnelle. Cette spécification est faite par l'urbaniste fonctionnel responsable de la zone en respectant la stratégie de l'entreprise, en particulier celle du marketing de l'entreprise. Le nombre d'interactions entre ces îlots fonctionnels permet de paramétrer une mesure évaluant la cohésion et le couplage de ces îlots fonctionnels de référence. La prise en compte de l'aspect dynamique permet ainsi de renforcer la stabilité dynamique du SI.

Un deuxième facteur de durabilité est la modélisation mise en œuvre comme solution à la complexité du SI d'une entreprise. La modélisation apparaît dans de nombreux items des démarches DA4EA et EA4UP. Elle est intégrée de façon classique dans la méta-modélisation des vues d'un système ou d'un SI. Les méta-modèles sont instanciés pour chaque modèle d'architectures conçu lors de l'application de ces démarches. Le choix de l'ingénierie des modèles et des transformations de modèle associées permet d'assister l'architecte en lui proposant une première version d'un modèle d'architecture. A l'aide de ce modèle initialisé, l'architecte peut faire des choix pertinents par rapport aux exigences de son système. Une originalité de l'application de l'approche MDA dans notre contribution est l'extension de la notion de PDM aux modèles d'EA, en particulier à celui de la vue fonctionnelle du SI. Ce modèle d'EA contraint les transformations de modèles

dont le résultat est un modèle d'architecture du système. Grâce à cette extension du PDM, les modèles faits par les architectes d'entreprise sont automatiquement intégrés lors du développement d'un système. Cette intégration des modèles de référence facilite la réutilisabilité d'éléments existants du SI et améliore ainsi sa durabilité.

La méta-modélisation des concepts des vues d'un SI ou d'un système permet aussi de définir un troisième facteur de durabilité d'un SI. Ce facteur est l'alignement entre certains concepts du méta-modèle de la vue métier et d'autres concepts de la vue fonctionnelle d'un SI dans la démarche DA4EA. Un constat intéressant de la pratique de cette mesure est l'apport pour l'urbaniste fonctionnel du bilan de cet alignement. Il permet en effet de mesurer l'écart entre la vue fonctionnelle du SI et les processus métier qui identifient l'entreprise. La question pour l'urbaniste fonctionnel du SI est de savoir si un meilleur alignement du fonctionnel sur le métier est ou n'est pas une bonne solution pour la cible fonctionnelle qu'il conçoit ? Le bilan d'alignement est donc une entrée utile à l'urbaniste fonctionnel lors de la conception inductive de sa zone fonctionnelle en fonction de la stratégie de l'entreprise. C'est cette réponse assistée de l'expert qui conditionne la durabilité du SI.

7. Perspectives

A court terme, la difficulté rencontrée par les projets pour mettre en œuvre la démarche EA4UP est sa complexité et son coût (cf. §4.4). Le coût élevé est principalement causé par l'appropriation de la vue fonctionnelle d'un SI par les architectes du système à développer.

Les retours sur la démarche DA4EA soulignent l'impact de l'automatisation, grâce à l'ingénierie des modèles, des tâches les plus complexes sur la simplification de la démarche et la diminution du coût. L'outillage de la démarche EA4UP est à envisager comme solution à la complexité de la démarche et à son optimisation en termes de coût.

La variation de l'information mutuelle moyenne indiquant l'amélioration de l'alignement fonctionnel entre deux itérations reste à évaluer sur des corpus d'interactions entre entités de taille suffisante. L'implémentation de la mesure est une condition nécessaire à cette évaluation. Pour les mêmes raisons de simplicité et de coût, la mesure d'alignement fondée sur l'information mutuelle moyenne (cf. §5.5) pourrait alors être intégrée à l'outillage de la démarche EA4UP.

De plus, la mesure d'alignement ne prend pas en compte l'estimation par les analystes du poids du scénario par rapport aux autres scénarios étudiés (par exemple, un scénario nominal *a*, *a priori*, un poids plus important qu'un scénario d'exception). Les probabilités calculées pour l'entropie de la variable aléatoire liée à la réception d'une interaction entre entités (cf. §5.5.2) pourraient intégrer une pondération de chaque scénario. Cette pondération est en effet classique dans un processus industriel de qualification ou de recette d'un système lors de la mesure de la qualité des exigences fonctionnelles réalisées.

Le taux de filtrage est fixé de façon empirique dans la démarche DA4EA. Une première remarque lors de ce travail empirique est l'influence de la taille du corpus de bactéries en entrée de l'algorithme (cf. §3.5.4). Toujours à court terme, il serait intéressant de compléter la démarche DA4EA par une définition de la relation entre le taux de filtrage et le taux de mutation de l'algorithme bactériologique.

A moyen terme, une perspective est d'enrichir le développement d'un système par les préconisations issues de l'urbanisme de la vue technique ou de l'urbanisme de la vue applicative. Cet enrichissement pourrait être concrétisé par une démarche et un outillage intégrant les notions de couches et de cadre architectural pour ces couches. La définition de couches pour une architecture N-tier est aussi une solution pour diminuer le couplage entre composants applicatifs d'un système [Shan, 2006]. La démarche DA4EA ainsi étendue à la vue technique et à la vue applicative pourrait aussi bénéficier de mesures d'alignement indiquant l'intégration des préconisations par les architectes de système.

A long terme, l'approche établie à partir de l'information mutuelle peut être utilisée pour d'autres processus que celui du développement de système. Pour ces autres processus, la seule nécessité est que l'approche dynamique soit un fondement des démarches qui accompagnent les processus. Cela peut être par exemple le cas du processus de qualification d'un système ou d'un logiciel [ISO – 12207]. La qualification fonctionnelle peut être complétée par une mesure d'alignement entre la vue applicative d'un système et la vue fonctionnelle qu'elle réalise. L'alignement serait fondé sur la cohérence des interactions entre composants applicatifs et des interactions entre îlots fonctionnels réalisant un scénario d'usage d'un système.

8. Références bibliographiques

- [Ambler, 1998] S.W. Ambler : "Process Patterns : Building Large-Scale Systems Using Object Technology". Cambridge University Press, 1998.
- [Ambler, 2005] S.W. Ambler, J. Nalbone, M.J. Vizdos : "The Enterprise Unified Process – Extending the Rational Unified Process". Prentice Hall – PTR, 2005.
- [Armour, 2001] F. Armour, G. Miller : "Advanced Use Case Modeling – Software Systems". Object Technology Series, Addison-Wesley, 2001.
- [Baldwin, 2000] C.Y. Baldwin, K.B. Clark : "Design Rules – The Power of Modularity". MIT Press, Cambridge, 2000.
- [Baudry1, 2005] B. Baudry, F. Fleurey, J.M. Jézéquel, Y. Le Traon : "From Genetic to Bacteriological Algorithms for Mutation-Based Testing". Software Testing, Verification and Reliability, 2005, vol. 15, pp. 73-96.
- [Baudry2, 2005] B. Baudry, F. Fleurey, J.M. Jézéquel, Y. Le Traon : "Automatic Test Case Optimization: A Bacteriologic Algorithm". IEEE Software, 2005, pp. 76-82.
- [Bertin, 2007] E. Bertin, I. Fodil, N. Crespi : "A business view for NGN service usage". The 2nd IEEE/IFIP International Workshop on Broadband Convergence Networks, München, Germany, 2007.
- [Blanc, 2005] X. Blanc : "MDA en action – Ingénierie logicielle guide par les modèles". Eyrolles, 2005.
- [Boehm, 1988] B. Boehm : "A Spiral Model of Software Development and Enhancement". IEEE Computer, 1988, pp. 61-72.
- [Booch, 1999] G. Booch, J. Rumbaugh, I. Jacobson : "The Unified Modeling Language – User Guide". Addison – Wesley, 1999.
- [Bonne, 2004] J.C. Bonne, A. Maddaloni : "Convaincre pour urbaniser le SI". Lavoisier, Publications Hermes – Science, 2004.
- [BRG, 2005] The Business Rules Group : "The Business Motivation Model". <http://www.businessrulesgroup.org/bmm.shtml>, 2005.
- [Briand, 1996] L.C. Briand, S. Morasca, V.R. Basili : "Property-Based Software Engineering Measurement". IEEE Transactions on Software Engineering, vol. 22, 1996, pp. 68-86.
- [Briand, 1999] L.C. Briand, J.W. Daly, J.K. Wüst : "A Unified Framework for Coupling Measurement in Object-Oriented Systems". IEEE Transactions on Software Engineering, vol. 25, No. 1, 1999, pp. 91-121.
- [Brien, 1997] J. O'Brien : "Les systèmes d'information de gestion". Ed. De Boeck Université, 1997.
- [Burlton, 2001] R.T. Burlton : "Business Process Management – Profiting from Process". Sams Publishing, 2001.
- [Cauvet, 2000] C. Cauvet, L. Gzara, P. Ramadour, D. Rieu : "Ingénierie des systèmes d'information produit : une approche centre réutilisation de patrons". Revue l'Objet, Hermès, vol. 6, n° 4, 2000, pp. 445-468.
- [Cauvet, 2008] C. Cauvet, G. Guzelian : "Business Process Modeling: a Service Oriented Approach". International Conference on System Sciences, Hawaï, 2008.
- [Chen, 2006] P.P. Chen : "Suggested Research Directions for a New Frontier – Active Conceptual Modeling". Conceptual Modeling – ER 2006, Springer, 2006.
- [Chidamber, 1994] S.R. Chidamber, C.F. Kemerer : "Towards a Metrics Suite for Object Oriented Design". IEEE Transactions on Software Engineering, vol. 20, No. 6, 1994, pp. 476-493.

- [Contini, 2002] I. Contini : "L'apport historique de l'urbanisme des villes pour l'urbanisme des systèmes d'information". Colloque Urbanisation des Systèmes d'Information, Paris, 2002.
- [Crozier, 1994] M. Crozier : "L'entreprise à l'écoute – Apprendre le management post-industriel". Editions du Seuil, 1994.
- [Cuevas, 2006] A. Cuevas, J.I. Moreno, P. Vidales, H. Einsiedler, "The IMS Service Platform: A Solution for Next-Generation Network Operators to Be More Than Bit Pipes". IEEE Communications Magazine, vol. 44, No. 8, 2006, pp. 75-81.
- [DoDAF, web] DoDAF : "Department of Defense Architecture Framework". http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf.
- [Drucker, 1983] P.F. Drucker : "The Concept of Corporation". New American Library, 1983.
- [Etien, 2006] A. Etien : "Ingénierie de l'alignement : Concepts, Modèles et Processus – La méthode ACEM pour l'alignement d'un système d'information aux processus d'entreprise". Thèse de Doctorat de l'Université Paris 1 – Panthéon – Sorbonne, 2006.
- [eTOM, web] TMForum : "enhanced Telecom Operations Map". <http://www.tmforum.org/browse.aspx?catID=1648>.
- [Fingar, 2000] P. Fingar : "Component-Based Frameworks for e-Commerce". Communications of the ACM, vol. 43, No. 10, 2000, pp. 61-66.
- [Fenton, 1986] N.E. Fenton, R.W. Whitty : "Axiomatic approach to Software Metrication through Program Decomposition". The Computer Journal, vol.29, 1986, pp. 330-339.
- [Fleurey, 2003] F. Fleurey, J. Steel, B. Baudry : "Validation in ModelDriven Engineering: Testing Model Transformations". Proceedings of the SIVOES – Modeva workshop, Rennes, 2004.
- [Frankel, 2003] D.S. Frankel : "Model Driven Architecture – Applying MDA to Enterprise Computing". Wiley Publishing Inc, 2003.
- [Gervais, 2002] M.P. Gervais : "Towards an MDA-Oriented Methodology". Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC'02), 2002.
- [Goldberg, 1989] D.E. Goldberg : "Genetic Algorithms in Search, Optimization and Machine Learning". Addison-Wesley, 1989.
- [Grigoriu, 2006] A. Grigoriu : "An Enterprise Architecture Development Framework", Trafford Publishing, 2006.
- [Grigoriu, 2007] A. Grigoriu : "SOA, BPM, EA, and Service Oriented Enterprise Architecture". BPTrends, www.bptrends.com, 2007.
- [Guelfi, 2004] N. Guelfi, G. Perrouin : "Using Model Transformation and Architectural Frameworks to Support the Software Development Process: the FIDJI Approach", Midwest Software Engineering Conference, Chicago, USA, 2004, pp. 13-22.
- [Hammer, 1993] M. Hammer, J. Champy : "Reengineering the Corporation: a Manifesto for Business Revolution. Harper – Collins, 1993.
- [Henderson, 1993] J.C. Henderson, N. Venkatraman : "Strategic Alignment: Leveraging Information Technology for Transforming Organizations". IBM Systems Journal, vol. 32, No. 1, 1993, pp. 4-16.
- [INPI – 04617] J. Simonin : "Modélisation UML et localisation des données référentielles". Dépôt de brevet France Télécom, dossier INPI 04617, 2003.
- [INPI – 05928] J. Simonin, M.F. Menai, F. Alizon : "Procédé de modélisation des données de contrôle réseau échangées sur un réseau de télécommunication sous contrainte de l'architecture des fonctions de ce réseau et des points de référence reliant ces fonctions ". Dépôt de brevet France Télécom, dossier INPI 05928, 2005.

- [INPI – 05929] J. Simonin, M.F. Menaï, F. Alizon : " Procédé de codage d'entités fonctionnelles réseau à partir de fonctions réseau d'un réseau de télécommunication et outil de modélisation correspondant". Dépôt de brevet France Télécom, dossier INPI 05929, 2005.
- [INPI – 06470] J. Simonin, J.P. Deschrevel, E. Bertin : "Procédé automatisé de conception d'un modèle des données dédiées à un service télécom". Dépôt de brevet France Télécom, dossier INPI 06470, 2006.
- [INPI – 06471] J. Simonin, J.P. Deschrevel, E. Bertin : "Procédé d'assistance automatisée à la conception d'un modèle arborescent des quartiers fonctionnels dédiés à un service télécom". Dépôt de brevet France Télécom, dossier INPI 06471, 2006.
- [ISO – 12207] International Organization of Standardization : "Software Life Cycle Processes". ISO/IEC 12207.
- [ISO – 15504] International Organization of Standardization : "Information technology – Process assessment". ISO/IEC PRF TR 15504.
- [ITIL, web] ITIL : "The IT Infrastructure Library". <http://www.itil-officialsite.com/home/home.asp>.
- [Jacobson, 1999] I. Jacobson, G. Booch, J. Rumbaugh : "The Unified Software Development Process". Addison-Wesley, 1999.
- [Jézéquel, 2008] J.M. Jézéquel : "Model driven design and aspect weaving". Software and Systems Modeling, Springer, vol. 7, No. 2, 2008, pp.209-218.
- [Jouault, 2006] F. Jouault, I. Kurtev : "On the Architectural Alignment of ATL and QVT". Proceedings of ACM Symposium on Applied Computing (SAC 06), Dijon, France, 2006.
- [Kaisler, 2005] S.H. Kaisler, F. Armour, M. Valivullah : "Enterprise Architecting : Critical Problems". International Conference on System Sciences, Hawaiï, 2005.
- [Kilov, 1994] H. Kilov, J. Ross : "Information Modeling – An Object-Oriented Approach". Prentice Hall, 1994.
- [Kitchenham, 1995] B. Kitchenham, S.L. Pfleeger, N. Fenton : "Towards a Framework for Software Measurement Validation". IEEE Transactions on Software Engineering, vol. 21, No. 12, 1995, pp. 929-943.
- [Kleppe, 2003] A. Kleppe, J. Warmer, W. Bast : "MDA Explained – The Model driven Architecture: Practise and Promise". Object Technology Series, Addison-Wesley, 2003.
- [Knightson, 2005] K. Knightson, N. Morita and T. Towle : "NGN Architecture: Generic Principles, Functional Architecture, and Implementation". IEEE Communications Magazine, vol. 43, Issue 10, 2005, pp. 49-56.
- [Kruchten, 1995] P. Kruchten : "Architectural Blueprints – The "4+1" View Model of Software Architecture". IEEE Software, vol. 12, No.6, 1995, pp. 42-50.
- [Larman, 2005] C. Larman : "Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design and Iterative Development". Prentice Hall, 2005.
- [Longépé, 2001] C. Longépé : "Projet d'urbanisation du système d'information – Démarche pratique avec cas concret". Dunod/01 Informatique, 2001.
- [Luftman, 1999] J.N. Luftman, R. Papp, T. Brier : "Enablers and Inhibitors of Business IT Alignment". Communications of the Association for Information Systems, vol. 1, Article 11, 1999.
- [MacDonell, 1997] S.G. MacDonell, M.J. Shepperd, P.J. Sallis : "Metrics for Database Systems: An Empirical Study". Software Metrics Symposium, 1997, pp. 99-107.
- [Markowitz, 1996] J.A. Markowitz : "Using Speech Recognition". Prentice-Hall, 1996.
- [MDA, web] OMG : "Model Driven Architecture (MDA)". <http://www.omg.org/mda/>.
- [Menaï, 2005] M.F. Menaï, G. Fromentoux, D. Gaïti, M. Lemercier : "A UML Tool for Urbanism and Control Architecture Design Applied to 3GPP Based Architectures". Proceedings of the international Tele-traffic Congress, vol. 6-b, 2005, pp 1987-1996.

- [Menaï, 2006] M.F. Menaï, J. Simonin, G. Fromentoux, D. Gaïti, M. Lemercier : "Urbanisation des architectures télécoms- stratégies pour le regroupement des fonctions en cohérence avec les données". Actes du congrès NOTERE, 2006, pp. 257-269.
- [Miller, 2003] J. Miller, J. Mukerji : "MDA Guide Version 1.0.1". <http://www.omg.org/docs/omg/03-06-01.pdf>.
- [Morin, 1990] E. Morin : "Communication et complexité – Introduction à la pensée complexe". ESF éditeur, 1990.
- [Muller, 2005] P.A. Muller, F. Fleurey, J.M. Jézéquel : "Weaving executability into object-oriented meta-languages". Proceedings of MODELS/UML'2005, LNCS, Montego Bay, Jamaica, 2005.
- [Nicolas, 2002] B. Nicolas : "MetYs". OMG Information Days, Paris, France, 2002.
- [OMA, 2005] Open Mobile Alliance : "OMA Service Environment". OMA-Service-Environment-V1_0_2-20050803-A.
- [Pereira, 2005] C. M. Pereira, P. Sousa : "Enterprise Architecture: Business and IT Alignment". ACM Symposium on Applied Computing, 2005, pp. 1344-1354.
- [PLU, 2006] Code de l'urbanisme : "Plans locaux d'urbanisme". <http://www.legifrance.gouv.fr/home.jsp>, Article L123-1, 2006.
- [QVT, web] OMG : "Query/View/Transformation (QVT)". <http://www.omg.org/docs/ptc/07-07-07.pdf>.
- [RM-ODP] RM-ODP. ITU-T Recommendation X.901 to X.904 | ISO/IEC 10746-1 to 4 and X.911 | ISO/IEC 15414, <http://www.rm-odp.net/>.
- [Robinson, 2004] R. Robinson : "Understand Enterprise Service Bus scenarios and solutions in Service-Oriented Architecture". <http://www-128.ibm.com/developerworks/webservices/library/ws-esbscen/index.html>.
- [Rolland, 1998] C. Rolland, C. Souveyet, C. Ben Achour : "Guiding Goal Modelling Using Scenarios". IEEE Transactions on Software Engineering, vol. 24, No. 12, 1998.
- [Rolland, 1999] C. Rolland, N. Prakash, A. Benjamin : "A Multi-Model View of Process Modelling". Requirements Engineering Journal, vol. 4, 1999, pp. 169–187.
- [Roques, 2004] P. Roques, F. Vallée : "UML2 en action – De l'analyse des besoins à la conception J2EE". Eyrolles, 2004.
- [Rosnay, 1975] J. de Rosnay : "Le macroscopie – Vers une vision globale". Edition du Seuil, 1975.
- [Rumbaugh, 1991] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen : "Object-Oriented Modeling and Design". Prentice-Hall International, 1991.
- [Salinesi, 2008] C. Salinesi, L.H. Thévenet : "Enterprise Architecture, des problèmes pratiques à l'innovation". Ingénierie des systèmes d'information, vol. 13, Lavoisier, 2008, pp. 75-105.
- [Sassoon, 1998] J. Sassoon : "Urbanisation des systèmes d'information". Hermès, 1998.
- [Schael, 1997] T. Schael : "Théorie et pratique du Workflow – Des processus métier renouvelés". Springer, 1997.
- [Shan, 2006] T.C. Shan, W.W. Hua : "Solution Architecture for N-Tier Applications". IEEE International Conference on Services Computing, SCC'06, Chicago, USA, 2006.
- [Shannon, 1948] C. Shannon : "A Mathematical Theory of Communication". Bell System Technical Journal, vol. 27, 1948, pp. 379-423, pp. 623-656.
- [Schekkerman, 2006] J. Schekkerman : "How to Survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework". Trafford Publishing, 2003.

- [Shepperd, 1993] M.J. Shepperd, D. Ince : "Derivation and Validation of Software Metrics". Oxford University Press, 1993.
- [Schulte, 1996] R.W. Schulte, Y.V. Natis : "Service Oriented Architectures – Part 1". Gartner, 1996.
- [Simonin, 2004] J. Simonin, F. Alizon : "Démarche d'urbanisation métier de type organisationnel". 17th International Conference Software and Systems Engineering and their Applications, Paris, France, 2004.
- [Simonin, 2007] J. Simonin, Y. Le Traon, J.M. Jézéquel : "An Enterprise Architecture Alignment Measure for Telecom Service Development". Proceedings of the 11th IEEE International EDOC Conference, Annapolis, USA, 2007, pp. 476-483.
- [Simonin, 2008] J. Simonin, F. Alizon, J.P. Deschrevel, Y. Le Traon, J.M. Jézéquel, B. Nicolas : "EA4UP: an Enterprise Architecture-Assisted Telecom Service Development Method". Proceedings of the 12th IEEE International EDOC Conference, München, Germany, 2008.
- [SmartQVT, web] G. Dupé, M. Belaunde : "SmartQVT". <http://smartqvt.elibel.tm.fr/>.
- [Tardieu, 1985] H. Tardieu, A. Rochfeld, R. Colletti, G. Panet, G. Vahée : "La méthode Merise – Tome 2 Démarches et pratiques". Editions d'organisation, Paris, 1985.
- [TOGAF, web] TOGAF : "The Open Group Architecture Framework". <http://www.opengroup.org/togaf/>.
- [UML, web] UML : "Unified Modelling Language". <http://www.omg.org/uml/>.
- [Urba-EA, 2006] Club Urba-EA : "Urbanisme des SI & gouvernance". Dunod, 2006.
- [Urba-SI, 2003] Club Urba-SI : "Pratique de l'urbanisme des systèmes d'information en entreprises". Publibook, 2003.
- [Wang, 2002] Y. Wang : "On Cognitive Informatics". Proceedings of the first IEEE International Conference on Cognitive Informatics ICCI'02, Calgary, Canada, 2002.
- [Wang, 2004] G. Wang, C.K. Fung : "Architecture Paradigms and Their Influences and Impacts on Component-Based Software Systems". International Conference on System Sciences, Hawaiï, 2004.
- [Warmer, 1999] J. Warmer, A. Kleppe : "The Object Constraint Language – Precise Modeling with UML". Object Technology Series, Addison-Wesley, 1999.
- [Wegmann, 2003] A. Wegmann : "On the Systemic Enterprise Architecture Methodology (SEAM)". International Conference on Enterprise Information Systems (ICEIS), Angers, France, 2003.
- [Wegmann, 2007] A. Wegmann, G. Regev, I. Rychkova, L.S. Lê, J.D. de la Cruz, P. Julia : "Business and IT Alignment with SEAM and Enterprise Architecture". Proceedings of the 11th IEEE International EDOC Conference, Annapolis, USA, 2007, pp. 111-121.
- [Wiseman, 1988] C. Wiseman : "Strategic Information Systems". Irwin, 1988.
- [Woodside, 2007] C.M. Woodside, G. Franks, D.C. Petriu : "Future of Software Performance Engineering". Future of Software Engineering 2007, L. Briand and A. Wolf (eds.), IEEE-CS Press, 2007, pp.171-187.
- [Zachman, 1987] J. A. Zachman : "A Framework for Information Systems Architecture". IBM Systems Journal 26, No. 3, pp. 276-292, 1987.
- [Zhang, 2000] X.J. Zhang, K.W. Ng : "An Effective High-level Synthesis Approach for Dynamically Reconfigurable Systems". The Fourth International Conference on High-Performance Computing in the Asia-Pacific Region, vol. 1, 2000.
- [Zimmermann, 1980] H. Zimmermann : "OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection". IEEE Transactions on Communications COM-28, No. 4, 1980

9. Liste des publications personnelles et des brevets

J. Simonin, F. Alizon, J.P. Deschrevel, Y. Le Traon, J.M. Jézéquel, B. Nicolas : "EA4UP: an Enterprise Architecture-Assisted Telecom Service Development Method". Proceedings of the 12th IEEE International EDOC Conference, München, Germany, 2008.

F. Alizon, M. Belaunde, G. Dupé, Y. Le Traon, B. Nicolas, S. Poivre, J. Simonin : "Les modèles dans l'action à France Télécom avec SmartQVT". Revue Génie Logiciel, n°85, juin 2008.

J. Simonin, Y. Le Traon, J.M. Jézéquel : "An Enterprise Architecture Alignment Measure for Telecom Service Development". Proceedings of the 11th IEEE International EDOC Conference, Annapolis, USA, 2007, pp. 476-483.

B. Nicolas, F. Alizon, M. Belaunde, G. Dupé, Y. Le Traon, S. Poivre, J. Simonin : "MDE experiments in France Telecom". Extrait de Shapping the Future : 10 years of Irisatech, Irisa, 2007, pp. 55-69.

M.F. Menaï, J. Simonin, G. Fromentoux, D. Gaïti, M. Lemerrier : "Urbanisation des architectures télécoms- stratégies pour le regroupement des fonctions en cohérence avec les données". Actes du congrès NOTERE 2006, pp. 257- 269.

J. Simonin, J.P. Deschrevel, E. Bertin : "Procédé d'assistance automatisée à la conception d'un modèle arborescent des quartiers fonctionnels dédiés à un service télécom". Dépôt de brevet France Télécom, dossier INPI 06471, 2006.

J. Simonin, J.P. Deschrevel, E. Bertin : "Procédé automatisé de conception d'un modèle des données dédiées à un service télécom". Dépôt de brevet France Télécom, dossier INPI 06470, 2006.

J. Simonin, M.F. Menaï, F. Alizon : " Procédé de codage d'entités fonctionnelles réseau à partir de fonctions réseau d'un réseau de télécommunication et outil de modélisation correspondant". Dépôt de brevet France Télécom, dossier INPI 05929, 2005.

J. Simonin, M.F. Menaï, F. Alizon : "Procédé de modélisation des données de contrôle réseau échangées sur un réseau de télécommunication sous contrainte de l'architecture des fonctions de ce réseau et des points de référence reliant ces fonctions ". Dépôt de brevet France Télécom, dossier INPI 05928, 2005.

J. Simonin, F. Alizon : "Démarche d'urbanisation métier de type organisationnel". 17th International Conference Software and Systems Engineering and their Applications, Paris, 2004.

J. Simonin : "Modélisation UML et localisation des données référentielles". Dépôt de brevet France Télécom, dossier INPI 04617, 2003.

L. Karray, J. Simonin : "A Trade-Off Between Recognition Performance and Computational Cost in Continuous HMM-Based Speech Recognition System". ASRU'99 IEEE Workshop, Keystone, Colorado, USA, 1999.

J. Simonin, L. Delphin-Poulat, G. Damnati : "Gaussian Density Tree Structure in a Multi-Gaussian HMM-Based Speech Recognition System". ICSLP'98, Sydney, Australie, 1998.

G. Damnati, J. Simonin : "Apport de l'Utilisation de Classes de Mots pour la Définition d'un Nouveau Lissage de type Absolute Discounting". JEP'98, Martigny, Suisse, 1998.

C. Mokbel, L. Mauuary, L. Karray, D. Jouvet, J. Monné, J. Simonin, K. Bartkova : "Towards Improving ASR Robustness for PSN & GSM Telephone Application". Speech Communication Journal, Vol. 23, Issue 1-2, 1997

J. Simonin, C. Mokbel : "Using Simulated Annealing Expectation Maximization Algorithm". EuroSpeech'97, Patras, Grèce, 1997.

G. Damnati, J. Simonin : "A Novel Tree-Based Clustering Algorithm for Statistical Language Modeling". EuroSpeech'97, Patras, Grèce, 1997.

M.D. Sadek, A. Ferrieux, A. Cozannet, P. Bretier, F. Panaget, J. Simonin : "Effective Human-Computer Cooperative Spoken Dialogue : the AGS Demonstrator". ISSD'96, Philadelphie, Pennsylvanie – USA, 1996.

J. Simonin, S. Bodin, D. Juvet, K. Bartkova : "Parameter Tying for Flexible Speech Recognition". ICSLP'96 , Philadelphie, Pennsylvanie – USA, 1996.

C. Mokbel, L. Mauuary, D. Juvet, J. Monné, C. Sorin, J. Simonin, K. Bartkova : "Towards Improving ASR Robustness for PSN & GSM Telephone Application", IVTTA 96, Philadelphie, Pennsylvanie – USA, 1996.

J. Simonin, S. Bodin, D. Juvet, K. Bartkova : "Combinaison de Différentes Modélisations Contextuelles pour la Reconnaissance de la Parole Flexible". JEP'96, Avignon, France, 1996.

10. Acronymes

2TUP	2 Track Unified Process
ATL	ATLAS Transformation Language
BPM	Business Process Management
CORBA	Common Object Request Broker Architecture
DA4EA	Dynamic Approach for Enterprise Architecture
DoDAF	Department of Defense Architecture Framework
EA	Enterprise Architecture
EA4UP	Enterprise Architecture for Unified Process
EUP	Enterprise Unified Process
eTOM	enhanced Telecom Operations Map
FEAF	Federal Enterprise Architecture Framework
IHM	Interface Homme - Machine
IS	Information System
ISO	International Organization for Standardization
ITIL	Information Technology Infrastructure Library
ITU	International Telecommunication Union
KerMeta	Metamodeling Kernel
MDA	Model Driven Architecture
MDE	Model Driven Engineering
MOA	Maîtrise d'OuvrAge
MOE	Maîtrise d'OEuvre
NGN	Next Generation Networks
OCL	Object Constraint Language
OMA	Open Mobile Alliance
OS	Operating System
OSI	Open Systems Interconnection
PDM	Platform Description Model
PIM	Platform Independent Model
POS	Plan d'Occupation des Sols
PSM	Platform Specific Model
PLU	Plan Local d'Urbanisme
RM-ODP	Reference Model for Open Distributed Processing
QVT	Query/View/Transformation
SI	Système d'Information
SOA	Service Oriented Architecture
SOEA	Service Oriented Enterprise Architecture
TMF	TeleManagement Forum
TOGAF	The Open Group Architecture Framework
ULS	Ultra-Large-Scale
UML	Unified Modeling Language
UP	Unified Process

11. Table des règles d'urbanisme

Règle d'urbanisme – Donnée produite et îlot fonctionnel n°1.....	28
Règle d'urbanisme – Donnée produite et composant applicatif n°2	28
Règle d'urbanisme – Interactions et voies n°3.....	43
Règle d'urbanisme – Flux et stock n°4	47
Règle d'urbanisme – Scission îlot multi-typé n°5	50
Règle d'urbanisme – Relation de dépendance entre données n°6.....	100
Règle d'urbanisme – Collecte des exigences fonctionnelles dynamiques n°7.....	109
Règle d'urbanisme – Analyse fonctionnelle statique n°8	112
Règle d'urbanisme – Analyse fonctionnelle dynamique n°9	114
Règle d'urbanisme – Analyse technique dynamique n°10.....	118
Règle d'urbanisme – Architecture fonctionnelle statique n°11.....	122
Règle d'urbanisme – Architecture fonctionnelle dynamique des îlots n°12.....	129
Règle d'urbanisme – Architecture fonctionnelle dynamique des données n°13.....	134
Règle d'urbanisme – Architecture applicative statique n°14 :	145
Règle d'urbanisme – Architecture applicative dynamique des composants n°15.....	149
Règle d'urbanisme – Architecture applicative dynamique des données n°16.....	154
Règle d'urbanisme – Alignement fonctionnel n°17	171

12. Table des Figures

Figure 1 : Processus liés à l'EA et rôles contributeurs	16
Figure 2 : Cadre de référence de Zachman	17
Figure 3 : Zone, quartier et îlot du PLU fonctionnel	19
Figure 4 : Îlot, parcelle et voie du PLU fonctionnel	20
Figure 5 : Illustration d'un diagramme d'activités UML	21
Figure 6 : Illustration d'un diagramme de classes UML	21
Figure 7 : Illustration d'un diagramme de séquence UML	22
Figure 8 : Illustration d'un stéréotype d'une classe UML	23
Figure 9 : Vues du cadre d'urbanisme d'un SI	24
Figure 10 : Méta-modèle de la vue fonctionnelle enrichie du PLU	25
Figure 11 : Règle d'urbanisme – Donnée produite et îlot fonctionnel	28
Figure 12 : UP et vues d'architecture	30
Figure 13 : Processus <i>se coucher dans un logement</i> et séquence ordonnée de services	32
Figure 14 : Processus <i>se coucher dans un logement</i> et dépendance de services	33
Figure 15 : PIM, PDM et PSM dans l'approche MDA	35
Figure 16 : Extrait de la vue métier eTOM	38
Figure 17 : Extrait de la vue fonctionnelle du SI de France Télécom	38
Figure 18 : Extrait de la vue fonctionnelle du SI des services télécom de France Télécom	39
Figure 19 : Extrait de la vue métier du SI des services télécom de France Télécom	39
Figure 20 : Démarche DA4EA	41
Figure 21 : Activité de conception experte des zones, des îlots et des voies fonctionnels dans la démarche DA4EA	42
Figure 22 : Modèle d'îlots fonctionnels de référence	43
Figure 23 : Diagramme de séquence conforme à un modèle d'îlots fonctionnels	44
Figure 24 : Illustration de la dépendance d'îlots de la vue fonctionnelle d'un SI	44
Figure 25 : Illustration d'un scénario instanciant des îlots dépendants de la vue fonctionnelle d'un SI	45
Figure 26 : Activité de typage des îlots fonctionnels dans la démarche DA4EA	46
Figure 27 : Règle de dépendance d'îlots fonctionnels suivant leur type	47
Figure 28 : Sous-activité de transformation des îlots fonctionnels par typage dans la démarche DA4EA	48
Figure 29 : Illustration de la scission d'un îlot fonctionnel respectant la contrainte par typage des îlots	49
Figure 30 : Sous-activité de transformation des voies fonctionnelles par typage dans la démarche DA4EA	50
Figure 31 : Illustration des îlots et des voies fonctionnels respectant la contrainte par typage des îlots	51
Figure 32 : Illustration d'un scénario instanciant des îlots respectant la contrainte par typage des îlots	52
Figure 33 : Activité de mesure de l'alignement avec la vue métier dans la démarche DA4EA	53
Figure 34 : Illustration d'une vue métier	55
Figure 35 : Activité de conception automatisée des quartiers dans la démarche DA4EA	60
Figure 36 : Illustration du couplage entre îlots fonctionnels à partir d'une approche dynamique de la vue fonctionnelle des îlots	61
Figure 37 : Procédure de conception automatisée des quartiers fonctionnels	73

Figure 38 : Extrait des îlots, des parcelles et des voies fonctionnels de la zone Messagerie conçus par l'expert	75
Figure 39 : Scénario d'envoi de message	76
Figure 40 : Scénario d'envoi d'un message vocal converti en MMS	77
Figure 41 : Scénario d'envoi d'un message vocal	78
Figure 42 : Scénario de réception d'un email infecté	78
Figure 43 : Scénario de notification par email d'une alerte de boîte de messagerie	79
Figure 44 : Scénario de notification par email de la lecture d'un email reçu	79
Figure 45 : Extrait des îlots et des voies fonctionnels de la zone messagerie contraints par le typage des îlots	81
Figure 46 : Scénario d'envoi d'un message vocal contraint par le typage des îlots	82
Figure 47 : Temps d'exécution de l'algorithme bactériologique par taux de mutation	86
Figure 48 : Nombre d'entités fonctionnelles générées par l'algorithme bactériologique par taux de mutation	87
Figure 49 : Qualité moyenne des entités fonctionnelles mémorisées par taux de mutation	88
Figure 50 : Qualité cumulée des entités fonctionnelles mémorisées par taux de mutation	89
Figure 51 : Activité de validation du PLU fonctionnel dans la démarche DA4EA	90
Figure 52 : Bilan de la démarche DA4EA	93
Figure 53 : Synthèse du bilan de la démarche DA4EA	95
Figure 54 : Vues du cadre de l'urbanisme, cœur de métier de l'entreprise, SI et systèmes	97
Figure 55 : Ordonnancement temporel et relation de dépendance dans un diagramme de séquence	99
Figure 56 : Vues d'architecture d'un système et démarche DA4EA	100
Figure 57 : Enchaînement de transformations de modèle	102
Figure 58 : Enchaînement de transformations de modèle, d'expert ou automatisable, dont l'une est conforme à un PDM	103
Figure 59 : Codage d'expert et transformation automatique	104
Figure 60 : Macro-activités de la démarche EA4UP	105
Figure 61 : Macro-activité de collecte des exigences dans la démarche EA4UP	108
Figure 62 : Activité de collecte des exigences fonctionnelles statiques	108
Figure 63 : Activité de collecte des exigences fonctionnelles dynamiques	109
Figure 64 : Concepts et transformation liés à l'activité de collecte des exigences fonctionnelles dynamiques	110
Figure 65 : Activité de collecte des exigences non fonctionnelles	110
Figure 66 : Macro-activité d'analyse dans la démarche EA4UP	111
Figure 67 : Activité d'analyse fonctionnelle statique	112
Figure 68 : Concepts et transformation liés à l'activité d'analyse fonctionnelle statique	112
Figure 69 : Illustration des entités de l'activité d'analyse fonctionnelle statique	113
Figure 70 : Activité d'analyse fonctionnelle dynamique	113
Figure 71 : Concepts et transformation liés à l'activité d'analyse fonctionnelle dynamique	114
Figure 72 : Illustration des interactions de l'activité d'analyse fonctionnelle dynamique	115
Figure 73 : Activité d'analyse technique statique	116
Figure 74 : Concepts et transformation liés à l'activité d'analyse technique statique	117
Figure 75 : Activité d'analyse technique dynamique	118
Figure 76 : Concepts et transformation liés à l'activité d'analyse technique dynamique	119

Figure 77 : Illustration des propriétés techniques dynamiques de l'activité d'analyse technique dynamique	120
Figure 78 : Macro-activité de conception de l'architecture fonctionnelle dans la démarche EA4UP	121
Figure 79 : Activité d'architecture fonctionnelle statique	122
Figure 80 : Concepts et transformation liés à l'activité d'architecture fonctionnelle statique.....	123
Figure 81 : Illustration des îlots fonctionnels de l'activité d'architecture fonctionnelle statique.....	125
Figure 82 : Activité d'architecture fonctionnelle dynamique des îlots.....	126
Figure 83 : Alignement d'interactions dans la règle d'urbanisme liée à l'architecture fonctionnelle dynamique des îlots – Codage	127
Figure 84 : Alignement d'interactions dans la règle d'urbanisme liée à l'architecture fonctionnelle dynamique des îlots – Alignement des interactions	128
Figure 85 : Concepts et transformation liés à l'activité d'architecture fonctionnelle dynamique des îlots	129
Figure 86 : Illustration des interactions entre îlots fonctionnels de l'activité d'architecture fonctionnelle dynamique des îlots	131
Figure 87 : Illustration des voies fonctionnelles de l'activité d'architecture fonctionnelle dynamique des îlots	132
Figure 88 : Activité d'architecture fonctionnelle dynamique des données	133
Figure 89 : Concepts et transformation liés à l'activité d'architecture fonctionnelle dynamique des données.....	134
Figure 90 : Illustration des données logiques de l'activité d'architecture fonctionnelle dynamique des données.....	136
Figure 91 : Macro-activité de conception de l'architecture technique dans la démarche EA4UP	138
Figure 92 : Activité d'architecture technique statique	139
Figure 93 : Concepts et transformation liés à l'activité d'architecture technique statique.....	140
Figure 94 : Illustration des nœuds d'exécution de l'activité d'architecture technique statique.....	140
Figure 95 : Activité d'architecture technique dynamique.....	141
Figure 96 : Concepts et transformation liés à l'activité d'architecture technique dynamique.....	142
Figure 97 : Illustration d'un protocole de l'activité d'architecture technique dynamique.....	142
Figure 98 : Activité d'architecture technique détaillée.....	143
Figure 99 : Macro-activité de conception de l'architecture applicative dans la démarche EA4UP	144
Figure 100 : Activité d'architecture applicative statique.....	145
Figure 101 : Concepts et transformation liés à l'activité d'architecture applicative statique	146
Figure 102 : Illustration de composants applicatifs de l'activité d'architecture applicative statique	146
Figure 103 : Illustration du lien de déploiement d'un composant applicatif de l'activité d'architecture applicative statique sur un nœud d'exécution	147
Figure 104 : Activité d'architecture applicative dynamique des composants ...	148
Figure 105 : Concepts et transformation liés à l'activité d'architecture applicative dynamique des composants.....	150
Figure 106 : Illustration des interactions entre composants applicatifs de l'activité d'architecture applicative dynamique des composants	151
Figure 107 : Illustration des interfaces applicatives de l'activité d'architecture applicative dynamique des composants.....	152
Figure 108 : Activité d'architecture applicative dynamique des données.....	153

Figure 109 : Concepts et transformation liés à l'activité d'architecture applicative dynamique des données	155
Figure 110 : Illustration des données physiques de l'activité d'architecture applicative dynamique des données	156
Figure 111 : Activité d'architecture applicative détaillée	158
Figure 112 : Macro-activité d'implémentation dans la démarche EA4UP	159
Figure 113 : Activité d'implémentation	159
Figure 114 : Macro-activité de test dans la démarche EA4UP	160
Figure 115 : Activité de test fonctionnel	161
Figure 116 : Activité de test de robustesse et de performance	161
Figure 117 : Activité liée au bilan du développement dans la démarche DA4EA	162
Figure 118 : Bilan de la démarche EA4UP	164
Figure 119 : Illustration d'une orchestration de services fonctionnels	166
Figure 120 : Illustration d'une orchestration de services applicatifs	167
Figure 121 : Synthèse du bilan de la démarche EA4UP	168
Figure 122 : Voie du PLU entre quartiers fonctionnels	170
Figure 123 : Voie du PLU entre zones fonctionnelles	170
Figure 124 : Alignement fonctionnel lors de l'activité d'architecture fonctionnelle dynamique des îlots	172
Figure 125 : Alignement fonctionnel et théorie de l'information	174
Figure 126 : Information mutuelle moyenne et entropie	175
Figure 127 : Analyse fonctionnelle dynamique et activité d'architecture fonctionnelle dynamique des îlots	176
Figure 128 : Contrainte du PLU fonctionnel et activité d'architecture fonctionnelle dynamique des îlots	177
Figure 129 : Système de communication	178
Figure 130 : Codage, décodage et activité d'architecture fonctionnelle dynamique des îlots	178
Figure 131 : Codage et approche MDA	179
Figure 132 : Illustration du codage de l'alignement fonctionnel	180
Figure 133 : Décodage et approche MDA	181
Figure 134 : Illustration du décodage de l'alignement fonctionnel	182
Figure 135 : Canal de communication et PLU fonctionnel	183
Figure 136 : Illustration du PLU fonctionnel	184

13. Table des Tableaux

Tableau 1 : Concepts de la vue métier.	25
Tableau 2 : Concepts de la vue fonctionnelle.....	26
Tableau 3 : Concepts de la vue technique.....	27
Tableau 4 : Concepts de la vue applicative.	27
Tableau 5 : Macro-activités UP et modélisation.....	31
Tableau 6 : Références des scénarios illustrant la zone Messagerie.	75
Tableau 7 : Alignement des parcelles fonctionnelles de la zone Messagerie avec les activités métier du processus d'envoi de message électronique.	83
Tableau 8 : Alignement des relations de dépendance entre parcelles fonctionnelles de la zone Messagerie avec les relations de succession entre activités métier du processus d'envoi de message électronique.	84
Tableau 9 : Réponses au questionnaire du bilan de la démarche DA4EA.	92
Tableau 10 : Alignement de la zone Messagerie "expert" et de la zone Messagerie DA4EA.	94
Tableau 11 : Macro-activité UP et activités EA4UP.	107
Tableau 12 : Illustration des propriétés techniques statiques de l'activité d'analyse technique statique.	117
Tableau 13 : Illustration du codage des éléments d'analyse fonctionnelle statique par les parcelles fonctionnelles du système.	124
Tableau 14 : Illustration du lien de réalisation d'une parcelle fonctionnelle par un composant applicatif de l'activité d'architecture applicative statique.....	147
Tableau 15 : Réponses au questionnaire du bilan de la démarche EA4UP.....	164